Molecular Recognition and Partner Prediction for Transient Protein Complexes: CDK-cyclin Homologue Interactions

Xueping Quan

A thesis submitted for the degree of Doctor of Philosophy

University of Edinburgh

July 2006

Declaration

This thesis has been composed by me myself from the results of my own work, except where stated otherwise, and has not been submitted in any other applications for a degree.

Acknowledgements

There are numerous people that contributed to the completion of this thesis. First and foremost, I wish to express my gratitude to my supervisors Dr. Dietlind L. Gerloff and Dr. Peter Doerner for their guidance, support and invaluable advice throughout the course of the work. I am also grateful to Professor Andrew Coulson, Dr. Alastair Kerr and Dr. Russell Hamilton for their help in thesis review; to Razif Gabdoulline & Rebecca Wade (EML & EMBL) for their allowance to use and modify MolSurfer; to Todd Dolinsky & Jens Erik Nielsen (WU St. Louis) for kind offer to use standalone PDB2PQR and their instructions; to Paul Taylor for the supply of extra machine to run large scale docking experiment; to my colleagues in ISMB department for their assistance in one way or another during the last three years.

I would like to thank the DARWIN Trust for providing the financial support without which I would not have been in the position to commence and complete this Ph.D. project.

Thanks are also extended to my colleagues in the Biocomputing Group for their assistance in one way or another during the last three years.

I wish also to thanks to my examiners Professor Malcolm Walkinshaw and Professor James Milner-White for their selfless examinations of this manuscript.

Table of Content

List of Figures

List of Tables

Acronyms and Abbreviations

2-D: two-dimensional

3-D: three-dimensional

aPK: atypical protein kinase

BLOSUM: BLOcks SUbstitution Matrix

CAPRI: Critical Assessment of Predicted Interactions

CASP: the Critical Assessment of Structure Prediction

CAFASP: the Critical Assessment of Fully Automated Structure Prediction

CDK: cyclin-dependent kinase

CEH: combination of ECC and HCC

ECC: electrostatic correlation coefficient

EP: electrostatic potential

ePK: eukaryotic protein kinase

ET: Evolutionary Trace

FFT: Fast Fourier Transform

HCC: hydrophobic correlation coefficient

HMM: Hidden Markov Model

HPRD: Human Protein Reference Database

LP: lipophilicity potential

PAM: Percentage of point accepted mutation per 108 years

PBE: Poisson-Boltzmann Equation

PIC: partition identity cut-off

RMSD: root mean square deviation

SASA: solvent accessible surface area

SCR: structurally conserved region

SIP: sequence identity percentage

STDV: standard deviation

SVR: structurally variable regions

1.

ABSTRACT

This thesis describes a novel computational strategy that combines multiple bioinformatics program components to predict specific transient protein-protein interactions. This protocol was developed focusing especially on the transient interactions between cyclin-dependent kinases (CDKs) and cyclins. To date a lot of bioinformatics research is undertaken on protein-protein interactions by various research groups. However, there are still no systematic computational methods available to predict specific protein-protein interactions between sets of paralogous proteins, e.g. which CDK-cyclin pairs can form transient complexes, and which do not. Here we adopted a comparative modeling strategy to build 3-D models of cyclins and CDKs using known human CDK and cyclin structures as templates. These modelled structures were then subjected to a large scale docking experiment with the program ZDOCK in which all cyclin-CDK combinations were considered. In the following steps of the procedure, additional selection criteria were applied to select the most compelling complexes from the ZDOCK result list. The two principal selection criteria used were the relative CDK-cyclin subunit orientation in the complex, and interface surface property correlation. Calibration of interface surface property correlation coefficients as computed by the program MOLSURFER was based on a positive reference dataset consisting of 104 true, non-homologous, transient heterodimeric protein-protein complexes, and a negative reference dataset consisting of 70 false protein-protein complexes. Prediction accuracies achieved using this approach are expected to be around 80% based on cross-validation of the interface selection criteria.

The entire modeling and prediction approach has been applied to the well-characterized set of human CDKs and cyclins. Of the resulting positive predictions, 80% were in agreement with complex formation according to HPRD and Swiss-Prot annotation. Finally, when the approach was applied to 33 CDK-homologues and 35 cyclin-homologues in *Arabidopsis thaliana* it yielded 19 mostly likely interacting CDK-cyclin pairs. The most strongly predicted complex is formed between a close homologue of human CDK1/2/3, and a sequence most similar to human cyclinA (human CDK1/2/3-cyclinA are natural pairs). Another predicted complex has recently been confirmed experimentally by another research group.

The prediction strategy developed and applied in this work should be transferable to other transient heterodimer protein-protein interactions.

2.

BACKGROUND

2.1 Protein Interactions with Other Molecules

Proteins are the most interesting and important of all molecules in biological systems. They are crucial to organisms (most organisms on earth except some viruses) which use them to carry out a huge variety of essential function for example catalysis, transport, storage, signaling, regulation, DNA unwinding and repair, etc. Proteins are made up from essentially combinations of amino acids in peptide linkages. A protein molecule that consists of a single polypeptide chain is said to be monomeric; proteins made up of more than one polypeptide chain, as many of the larger ones are, are termed oligomeric. Based upon chemical composition, proteins are divided into two major classes: simple proteins, yielding only amino acids when hydrolyzed, and conjugated proteins, complexes combining amino acids with other substances. Conjugated proteins include glycoproteins containing carbohydrates components; lipoproteins containing lipid components and are principal means for transporting lipids in the blood; and nucleoproteins containing nucleic acids. Classified by biological function, proteins include the enzymes, "biological catalysts" responsible for increasing rates of thousands of chemical reactions in living cell; structural proteins, for example elastin and collagen; haemoglobin, a conjugated protein linked to an iron-porphyrin compound, and other gas transport proteins; ovalbumin, casein, and other nutrient molecules; antibodies, Y-shaped proteins secreted into blood or lymph in response to antigenic stimulus; protein hormones, which regulate metabolism; and proteins that perform mechanical work, such as actin and myosin, the contractile muscle proteins (the Columbia Encylopedia, 6th Edition, 2001-2005).

Proteins need to fold into their native shape to be able to function normally. Usually less than one minute after proteins are formed, the linear peptide chains fold into their pre-ordain shape. Mis-folding can affect the functions of proteins severely. A typical example is the prion protein, whose normal fold, PrP^{C}, consists of mainly α-helices. Through conformational change, prions mis-fold into the PrP^{Sc} shape state consisting of a substantial proportion of β−sheet structures. Prion proteins with the PrP^{C} conformation function normally as part of living cells, the mis-folded prions with the PrP^{Sc} conformation cause diseases to animal brains (Horwich AL and Weissman JS, 1997).

Most proteins' functions depend on interaction with other molecules, including other proteins, nucleic acids, solvent molecules, metal ions, ligands, etc. Protein interactions operate at almost every level of cell function. Processes as diverse as cytoskeletal remodelling, vesicle transport and signal transduction, packaging of chromatin, the network of sub-membrane filaments, muscle contraction, regulation of gene expression, regulation of cell cycle, to name a few, are all dependent on physical interactions between proteins and other molecules. For example, the function of cyclins, a family of α-helical proteins involved in the regulation of cell cycle, function involves reversible binding of another family of proteins: cyclin-dependent kinases (CDKs). DNA helicase, an important protein for the repair of damaged DNA, mainly facilitates the splitting apart of the two strands of the DNA double helix. In order to do this, it not only has to bind DNA to break the hydrogen bonds between the DNA strands, but must also bind ATP to obtain energy to perform this function.

There are several different models to explain interactions between proteins and other molecules. For example, in 1894 Emil Fischer (Fischer EE, 1894) wrote that an enzyme and its substrate, usually a small molecule, fit together like a "lock" and "key". The active site of an enzyme usually has a unique geometric shape that is complementary to the geometric shape of the substrate molecule(s) and sometimes co-factors. In this analogy, the lock is the enzyme and the key(s) are the substrate and co-factor molecules. Most enzymes are therefore very specific; they will only function correctly if the shapes of the substrate and/or co-factor molecules match the active site. Not all experimental evidence can be adequately explained by using the so-called rigid enzyme model assumed by the lock and key theory. For this reason, a modification called the induced-fit theory was proposed (Koshland DE Jr. 1958). The induced-fit theory assumes that the substrate may play a role in determining the final shape of the enzyme and that the enzyme is partially flexible. This explains why certain compounds can bind to the enzyme but do not lead to a reaction because the necessary conformational change (induced fit) does not occur. Structural plasticity is also evident in interactions of proteins with other molecules. Spatial adaptations of several antibody-antigen complexes have been demonstrated by high-resolution crystal structure analysis, for example (Davis DR & Cohen GH, 1996).

The pre-existing equilibrium hypothesis (Tsai CJ, et al, 1999) is based on the protein folding theories of the funnel energy landscape. Instead of making simplistic assumptions as in the "lock-key" and "induced-fit" models, that a protein has a global minimum energy corresponding to the existence of a single structural conformer, the pre-existing equilibrium hypothesis assumes that proteins have an energy landscape with many local minima corresponding to an ensemble of pre-

existing conformations with similar but discrete energy levels. The binding of partners (substrates and co-factors) biases the equilibrium toward protein molecules in the binding conformation. For example, the ester-hydrolysing antibodies, D2.3, D2.4 and D2.5 (Linder AB, et al, 1999), were originally assumed to comply with the rigid "lock-key" model with their bound and un-bound structures. However, pre-steady-state kinetics revealed a pre-equilibrium between two antibody isomers, only one of which binds the hapten with high affinity.

Protein-protein interactions, the subject of this chapter, often involve conformational changes and cannot be explained by the "lock-key" theory. The "induced fit" model postulates that a protein substrate (peptide) molecule can induce a conformational change at the interaction site of the enzyme protein upon binding and would thus explain protein-protein interactions with "local" conformation changes. Tobi and Bahar (Tobi D & Bahar I, 2005) investigated the equilibrium motions of proteins exhibiting relatively large (non-local) conformational changes on bindings. Their study found that proteins, in their native conformation, are predisposed to undergo conformational changes that are relevant to their functions and the "pre-existing equilibrium" mechanism was proposed for these kinds of protein-protein interactions.

2.2 Protein-Protein Interaction Classification

Protein-protein interactions can be found to form homo- or hetero- complexes based on whether the interaction occurs between identical or non-identical chains. An enormous number of proteins, including many enzymes, carrier proteins, and characterized structural proteins function as homo-oligomers. Klotz et al analyzed ~300 protein entries (primarily soluble enzymes) presumed to form oligomers (Klotz

IM, *et al*, 1975). Over half of the oligomeric proteins were homo-dimer or homo-tetramers, and only ~15% were hetero-oligomers made of different chains. A more recent survey in May.2006 (Thomas Juettemann unpublished result) found that 14641 out of 35315 pdb entries containing multiple protein chains were pure homo-oligomers. Homo-oligomers are often (but not always, e.g in trimers) symmetrical, and comprise even numbers of subunits. Symmetrical oligomers are favoured due to stability and finite control of assembly. Oligomerisation can faciliate ligand binding or modify the protein conformation in response to regulatory ligands. In this sense, oligomerisation may also make the activity of certain proteins dependent on protein concentration. This provides a means for controlling function either positively or negatively, depending on whether the oligomer or the monomer exhibits the highest activity.

Hetero-interactions are largely responsible for the transduction of physical or chemical information signals for communication at the level of the cell or the organism. They play an important role in the control of cell growth, differentiation and development. Hetero-interactions often involve specific interactions between domains. For example, Src homology domains 2 and 3 (SH2 and SH3) are found in growth factor receptor-binding proteins involved in signal transduction; the basic helix-loop-helix (bHLH) domains of proteins TAL1, TAL2 and LYL1 interact with the cysteine-rich LIM (named from the Lin-11, Isl-1 and Mec-3 genes) domains of proteins RBTN1 and RBTN2 that are involved in transcriptional regulation (Wadman I, *et al*, 1994).

Based on whether a complex is "obligate" or "non-obligate" (Jones S & Thornton JM, 1996), protein-protein interactions (PPI) can be classified to produce permanent or

transient complexes. The monomers of a permanent complex cannot be found as stable structures on their own *in vivo*. Obligate interactions are usually very stable. Many proteins that exist as parts of permanent obligate complexes such as multi-subunit enzymes, fold and bind simultaneously.

Transient interactions are postulated to control the majority of cellular processes with interacting partners associating and dissociating *in vivo*. Transient complexes are therefore often unstable and may be difficult to purify and crystallise. Examples include complexes formed through enzyme-inhibitor, enzyme-substrate, hormone-receptor, and signalling-effector types of interactions.

2.3 Principles of Protein-Protein Interactions

2.3.1 Protein-Protein Contact Area

One of the main features of a protein-protein recognition site is the interface area: the area of the accessible surface on both partners that becomes inaccessible to solvent on protein-protein binding, formally termed as the change in their solvent accessible surface area (ΔASA) upon complex formation:

$$\Delta ASA = 1/2\ (SASAa + SASAb - SASAab)$$

Where a and b are the two protomers in the complex "ab"; SASAa, SASAb and SASAab, are the SASA (solvent accessible surface area) value for a, b and ab, respectively.

The ΔASA on complex formation of homo-dimers varies widely from 360 $Å^2$ to 4800 $Å^2$ for single subunits (Jones S. & Thornton J.M., 1996). The variation of ΔASA for hetero-complexes seems to be smaller as it ranges from 630 $Å^2$ to 3228 $Å^2$ for each subunit. There seems to be an obvious difference between transient homo-

dimers whose interface ΔASA ranges from 470 Å^2 to 930 Å^2 and transient hetero-dimers whose ΔASA vary from 570 Å^2 to 2220 Å^2 (Nooren IMA and Thornton JM, 2003).

The contact area difference may also be useful for discriminating between different experimentally determined structures, including crystallographic structures, NMR structures, and models by homology (Abagyan RA and Totrov MM, 1997).

2.3.2 Forces that Mediate Protein-Protein Interactions

There are various physical and chemical forces between two interacting molecules. Fundamental to the stabilisation of protein association are hydrophobic interactions formed between non-polar groups (Chothia C and Janin J, 1975). A hydrophobic interface will drive the formation of complexes as hydrophobic residues aggregate away from contact with water as hydrophobic surfaces tend to be more "de-wetting" than hydrophilic surfaces (Jensen MO, et al, 2004; Dill KA, et al, 2005). Hydrophobic forces mainly come though van der Waals contacts between non-polar regions of their amino acid residues. Van Der Waals forces occur between all proximal atoms, and the interactions at the interface are no more energetically favourable than those made with the solvent. However, they are more numerous, as the tightly packed interfaces are denser than the solvent shell. Van der Waals forces are contributing when the two molecules are close and their contact surface is large (http://www.nanomedicine.com/NMI/3.5.1.htm). The following formula is used to determine the Van der Waals bonding between two parallel plates of area A:

$$E_{vdW} = HA/12\pi Z^2$$

Here Z is the distance between the two planes. H is Hamaker constant. H = 37zJ for water, 66zJ for glycerol.

Desolvation (the expulsion of surrounding water) from hydrophobic part of protein surfaces that interact in the complex provide the main driving force for complex formation. The desolvation of charged residues during interaction is destabilising at the first instance and the hydrophobic effect described above is typically viewed to be the primary driving force for complex formation. However, charged groups located at an interface can often be stabilised by other polar and oppositely charged groups on the interacting partner molecule. Electrostatic forces are the strongest force that draws parts of the molecules closer together or pushes them further apart, depending on their electric charge. The potential of hydrogen bond formation can strengthen the bond between two molecules substantially and occurs when one molecule has a hydrogen bond donor close to the contacting surface that interacts with a hydrogen bond acceptor from the second molecule when interaction occurs. Salt bridges help to stabilise protein structures and protein-protein associations by limiting the number of low free energy conformations and by charge neutralization (Hendsch ZS and Tidor B, 1994). Salt bridges also have a tendency to form additional hydrogen bonds in proteins (Honig B and Hebbell WL, 1984).

Overall, the polar interaction forces must be stronger than the desolvation penalty associated with the burial of polar or charged residues at the protein-protein interface (Janin J *et al*, 1988; Chothia C and Janin J, 1975). Interfaces in obligate associations tend to have fewer hydrogen bonds than interfaces in non-obligate associations (Uetz P & Vollert CS, http://iqtmv1.fzk.de/www/itg/uetz/publications/Uetz2003-PPI.pdf.). Therefore charge complementarity, i.e., electrostatic potential, including hydrogen

bonds and salt bridges, is likely to also play an important role in determining the specificity of the interaction.

Overall protein-protein interfaces are not less polar (more hydrophobic) than the surfaces remaining in contact with the solvent (Jones S and Thornton JM, 1996; Janin J, *et al*, 1988). Obligate complexes on average seem to have interfaces that are slightly more hydrophobic than the rest of the molecular surface. Non-obligate complexes tend to be more hydrophilic in comparison, presumably because each component has to exist independently in the cell (Uetz P & Vollert CS, http://igtmv1.fzk.de/www/itg/uetz/publications/Uetz2003-PPI.pdf.).

2.3.3 Shape and Shape Complementarity

Favourable interactions between proteins generally require the molecules to fit well together spatially because bumps and clashes would cause energetic repulsion. Two independent surveys showed that around 84% of interfaces are essentially flat (Jones S & Thornton JM, 1995; Argos P, 1988). With a few exceptions, the interfaces are approximately circular areas on the protein surface in both obligate and non-obligate complexes. Interfaces in obligate associations tend to be larger, less planar, more highly segmented (in terms of sequence), and closer packed than interfaces in non-obligate associations (Jones S & Thornton J, 1996).

The formation of protein-protein complexes benefits from an optimal correspondence of complementary residues at their interface. Presumably, the protein shape complementarity observed in contemporary protein complexes have evolved to optimise geometric compatibility as well as associated electrostatic forces at the protein-protein interface. Shape complementarity has been quantified indirectly in

terms of buried (solvent inaccessible) surface area (Chothia C., 1976), gap volume (Jones S & Thornton JM, 1995; Laskowski RA, 1991) and atomic packing density (Richards FM, 1974; Chothia C and Janin J, 1975; Hubbard SJ & Argos P, 1994). Solvent inaccessible surface area mentioned above will be explained in more detail later in the chapter 2.6. Gap volume (Laskowski RA, 1991) gives a measure of the complementarity of the interacting surfaces by estimating the volume enclosed between any two molecules. The first step in the calculation of gap volume between two subunits involves selecting a pair of atoms located on the interface around the gap, placing a sphere between these two atoms so that its surface touches the surface of the atoms in each of the pair. The radius of the sphere is usually defined to be between 0.5Å and 5Å. The maximum of the sphere radius will influence the maximum size of the gap region and the maximum distance of the gap margin to the protein (SURFNET (LasKowski , 1991) parameter definition). This step is repeated untill all pairs of atoms on the interface are considered. The sizes of all the permitted spheres are used to calculate gap volume. Gap volume index is defined as:

Gap Volume Index = Gap Volume / Interface ASA.

The number and size of protein cavities, "holes" or packing defects in/between protein tertiary structures that are entirely surrounded by protein atoms, and the nature of surrounding residues can provide useful guidelines for protein modelling and design. Water-filled cavities generally involve more polar surface regions and are typically larger. Their constituent water molecules are used to satisfy the local hydrogen bonding potentials (Hubbard SJ & Argos P, 1994). Inter-subunit and inter-domain cavities (cavities with atomic surface components coming from more than one subunit, and from more than one domain, inter-domain cavities) occupy a

significant fraction of their interfacial surfaces, are (on average) larger than intra-domain cavities (cavities taken from single domains) and more frequently water-filled (Chothia C & Janin J, 1975; Hubbard SJ & Argos P, 1994).

Recently, more direct methods have been developed (Lawrence MC & Colman PM, 1993; Norel R, *et al.*, 1994) comparing computer generated molecular surfaces. Many of these methods have been developed as part of docking algorithms which will be described in chapter 2.5. Interfaces in homo-dimers, enzyme-inhibitor complexes, and obligate hetero-complexes display the most shape complementarity, whilst the antibody-antigen complexes and the non-obligate hetero-complexes display the least shape complementarity (Jones S & Thornton JM, 1996).

2.3.4 Amino Acid Composition and Secondary Structure

In terms of their amino acid composition, interfaces show a greater similarity to the exterior of proteins than the interior. The hydrophobicity of the average interface in a multi-meric protein lies between that of the exterior and the interior (Jones S and Thornton JM, 1995; Korn AP and Burnett RM, 1991; Argos P, 1988). In Jones and Thornton's study, charged and polar residues, especially arginine and asparagines, appear at interfaces more frequently and the hydrophobic residues methionine and proline appear slightly less frequently. In general interfaces appear to have a preference for aromatic amino acids; two of the three aromatic residues are found preferentially at interfaces. This might suggest that aromatics make particularly good 'glue' for sticking protein subunits together.

In Jones and Thornton's study (involving only 28 homo-dimers) the authors also calculated the number of interface residues in each type of secondary structural

conformation as a percentage of the total number of interface residues (Jones S and Thornton JM, 1995). 53% of the interface residues were α-helical, 22% in extended (β-strand) conformation, and 12% being αβ. This result is comparable to Argos' (Argos P, 1988) who investigated the distribution of secondary structural states at interfaces according to their surface contribution. Argos concluded that loop (i.e. regions that are neither α-helices nor β-strands) interactions contributed on average about 40% of the interface contacts.

2.3.5 Hot Spots on Protein-Protein Interfaces

It is sometimes assumed that the energy of protein-protein binding is directly related to the buried hydrophobic surface area (Chothia & Janin, 1975; Horton & Lewis, 1992; Jones & Thornton, 1996). However, in their study of 12 protein hetero-dimers, at the level of side-chains, Bogan and Thorn find only low correlation between buried surface area and energy but a highly uneven distribution of energetic contributions by individual residues across each interface, and that certain residues seemed responsible for the bulk of the binding energy (Bogan AA and Thorn KS, 1998; Clackson T and Wells JA, 1995). Based on Alanine scanning mutagenesis and thermodynamic calculations, they found that the residues that contribute a large amount of binding energy (>3.5kcal/mol) tend to cluster together over a small area (~600 $Å^2$). They described such areas as "hot spots" and noted that they are located near the centre of the interfaces in the complexes they studied. "Hot spots" were aligned in both partner proteins and enriched in tryptophan, tyrosine and arginine residues. Surrounding them were usually energetically less important residues most likely serving to occlude bulk solvent from the centre. It has recently been shown

that some "hot spots" of protein interaction show a tendency for interactions with a variety of partners (Delano *et al.*, 2000).

2.3.6 Specificity of Protein-Protein Interactions

Cells are very complex and crowded compartments and proteins are generally enclosed with many potential binding partners with different surface properties. Understanding specificity in protein-protein recognition is the key to understanding the plethora of interaction networks in cells. Some proteins are very specific in their choice of binding partner (Nooren IMA and Thornton JM, 2002), for example in a signal transduction pathway a protein may have to bind strongly to one other protein to trigger the appropriate response, and not bind to any of a multitude other proteins to avoid triggering inappropriate responses. Some proteins are multi-specific, and can interact with several different interaction partners on coinciding or overlapping interfaces. Multi-specificity of protein-protein interaction can be distinguished to involve either multiple partners of the same protein family (as in the interactions between a CDK (cyclin-dependent kinase) and several cyclins which all also occur on the same interface), or a set of non-homologous proteins (as in the interactions between a CDK, a cyclin and a CDK substrate protein).

Some interactions are likely to be mutually exclusive, resulting in competition between alternative interaction partners for complex formation. The relative efficiency of complex formation with alternate interaction partners within the cell are determined by their intrinsic binding affinities, levels of expression, sub-cellular distribution, interactions with other partners or scaffolds, and many other factors in the cell. Among these factors, the local concentration of competing proteins and the

affinity for the target protein are clearly important. Binding affinity, the forces between proteins that cause them to combine, mainly derives from shape complementarity and chemistry determining the free energy of binding (Nooren IMA and Thornton JM, 2002).

2.4 Computational Approaches to Protein-Protein Interaction Prediction

Experimental approaches like yeast-two-hybrid screens, co-immune-precipitation, X-ray crystallography and NMR spectroscopy, can be applied to detect interacting protein pairs. However, some of these experimental approaches, X-ray crystallography and NMR spectroscopy especially, are expensive and time-consuming, and the other approaches, for example yeast-two-hybrid screen, have the problem that they may not provide unequivocal evidence in support/against two proteins interacting. Therefore computational approaches have been developed to predict possible protein-protein interactions.

Methods for protein-protein interaction prediction mainly try to answer one, or several, of three questions. First of all there is a principal question: whether this protein's function is to interact with other proteins or not. The general property is quite well conserved during evolution and so homologues of proteins that are involved in protein-protein interactions usually will also interact with proteins (Tatusov, *et al.* 1997; Andrade, *et al.* 1999; Pellegrini, *et al.* 1999). Accordingly, prediction approaches typically look for sequence similarities. The popular sequence search programs BLAST (Altschul SF, *et al.,* 1990; Gish W and States DJ, 1993; States DJ and Gish W, 1994; Karlin S and Altschul SF, 1993), FASTA (Lipman and Pearson, 1985; Pearson and Lipman, 1988), SSEARCH (Smith TF & Waterman MS,

1981), PSI-BLAST (Altschul SF, *et al.*, 1997), and various HMM-based programs, for example HMMER (Eddy SR, 1998), SAM(Hughey R & Krogh A, 1996; Karplus K et al, 1998), and META_MEME (Grundy *et al*, 1997), are often used to find similar protein or nucleic acid sequences in a sequence database. The accuracy level with which functional attributes can be inferred between similar sequences depends on the evolutionary distance between those sequences. Sequences (larger than 100 amino acids) with sequence percentage identity higher than 35% (PAM value between 20 and 160) are very likely homologous (Sander and Schneider, 1991). Alternatively, one can examine the surfaces of folded proteins if structures are known and look for similarity that might indicate similar properties. For example, the program PIPSA by Rebecca Wade's group was designed to do this automatically (Blomberg N., *et al*, 1999). In PIPSA, protein electrostatic potentials, quantitatively represented as similarity indices, are calculated to compare a set of superimposed proteins. A matrix of pair-wise similarity indices for each protein electrostatic potential are then converted into a distance matrix which can be used for protein clustering and visualization. The Hodgkin index is commonly used to measure the similarity of two molecular potentials, detecting differences in charge, magnitude, and spatial behaviour in the potentials.

The next questions are more specific. One can try to predict the binding site of an interacting protein even if one does not know exactly which protein it is, or predict the interacting partner protein, or, of course, both.

In cases where the location of binding sites are conserved it has proven useful to analyse multiple sequence alignments and look for sequence conservation in subfamilies. Particularly where a family of homologous proteins maintain a specific

binding partner, this is also reflected in sequence conservation to a higher degree at the interface when compared to other regions on the monomeric surfaces. Very generally, functional sites undergo fewer mutations during evolution than other parts of protein as functional sites are under natural selection pressure to maintain their functional integrity (Zvelebil *et al.*, 1987; Ludwig M, *et al*, 2000; Berg J *et al*, 2004). Conversely, conservation of residues at the surface of protein families are usually indicative of their functional relevance. Sites containing conserved residues could be functional sites such as enzyme active sites or protein binding sites. In order to infer structure-function relationships based on conservation, the proteins must be evolutionary (and therefore structurally) related. Ideal targets for such analyse are large families of proteins with related function.

There are various programmes that do this. The most successful is the Evolutionary Trace (ET) method by the Lichtarge group (Lichtarge *et al,* 1996). It has been shown to work well in cases of SH2 domains, zinc fingers and other examples. Generally these methods will benefit from knowing the 3-D structures. In the case of the ET method, the structure of at least one member of the large family is required for analysis.

Knowing the structure of one binding partner can help a little but does give extra clues. For example, some domain folds always interact with proteins at the same side. Evidently, super-families with multiple partners (domain-combinations with two or more superfamily domains) vary their orientation and the region of the surface involved in domain interactions to a greater degree than super-families which have only one domain type partners (Littler and Hubbard, 2005). One can also examine surface properties for example shape, hydrophobic and electrostatic properties, in

more detail (Laskowski *et al* 1996; Jones & Thornton, 1997). Protein-protein interfaces have been observed to be planar, globular, and hydrophobic. However, there are no general methods that utilise these trends systematically for prediction till now. One can try using molecular docking programs to predict binding sites but the structures of both partners in an interacting pair will have to be known for this.

The third interesting problem is to predict the interacting partner. One can either ask this question categorically and make a general statement about which kinds of proteins are likely to interact with each other, for example, the CDK family and the cyclin family. At the structural end, work in Teichmann group has shown that certain domain folds prefer to interact with other specific domain folds, probably because of evolutionary conservation of protein-protein interactions and this can help prediction (Park *et al.*, 2001). In addition, there are a number of indirect approaches that can be applied to whole genomes and take advantage of interspecies sequence comparisons. For example conservation of gene neighbourhood in bacteria usually indicates functionally related proteins and some of these may interact with each other; gene fusion events of orthologous protein domains in different genomes which either form part of single polypeptide chain or are produced as separate chains; greater degree of similarity between interacting proteins' phylogenetic profiles than between those of non-interacting proteins (Pellegrini *et al*, 1999; Gaasterland T *et al* 1998; Tamames *et al*, 1997; Dandekar *et al* 1998; Overbeek *et al* , 1999;Marcotte *et al*, 1999; Enright *et al*, 1999; Tsoka S, *et al*, 2000; Sprinzak and Margalit, 2001; Fryxell, 1996; Pages S *et al*, 1997; Goh *et al*, 2000; Pazos and Valencia, 2001).

However, none of these approaches are applicable when one tries to predict which specific protein within a set of close homologues interacts with which protein in

another set of homologues, for example, which CDK is likely to interact speciafically with which cyclin. Only molecular docking approaches with individual protein structures appear suitable for attempting to answer this question (Karchalski-Katzir *et al.* 1992; Walls and Sternberg, 1992).

2.5 Molecular Docking

The application of computational methods to study the formation of intermolecular complexes has been the subject of intensive research during the last decade. Docking is a modelling process in which the interactions between molecules are evaluated computationally. Docking algorithms aim to model complexes between macromolecules (*e.g.* protein-protein or protein-DNA) or between a macromolecule and a small molecule-macromolecule, which is most common in drug design. Accordingly, two groups of docking programs can be distinguished: protein-protein docking and protein-ligand docking programs. The docking process, generally, involves dealing with issues relating to conformational flexibility, search algorithm, and score function.

2.5.1 Conformational Flexibility

It is not clear to what extent proteins change their conformation upon forming a complex. However, it is possible to distinguish various levels of conformational change: no change, side chain movements alone, segment movement involving the main chain, domain movements, and even more drastically changing from "disordered" to "ordered" (Halperin I, *et al*, 2002). Based on the extent of flexibility that the functions inherent in docking algorithms attempt to address, they can be classified into three levels (Fraga S, *et al*, 1995): 1) Rigid body docking treating the

two proteins as rigid solid bodies, but allowing a certain extent of surface variability; 2) Semi-flexible docking that regards one molecule, usually the smaller ligand, as flexible, and the other molecule, receptor, as rigid; 3) Flexible docking that considers both molecules as flexible, although the extent of flexibility of either or both of the two molecules needs to be limited or simplified. However, only two classes are distinguished by most researchers in the field: rigid body docking and flexible docking (here including semi-flexible and flexible docking). Since ligands are small-sized and are therefore likely to undergo larger conformational fluctuations during interaction, protein-ligand docking is usually accomplished through flexible docking. The small size of one interaction partner also makes it computationally affordable to run flexible docking in protein-ligand studies. By contrast, protein-protein docking is usually attempted through rigid body docking as the large partner sizes renders flexible docking computationally unaffordable. Most method development in docking has been, and continues to be, targeted towards those complexes for which the conformational changes upon docking are fairly small, thus enabling the use of methods whose primary attention is on shape and chemical complementarity of the unbound components (Smith GR and Sternberg MJE, 2002). The advantage of such rigid body docking programs in practice is that their speed enables their application to screening problems without requiring high performance computing. However, complexes modelled through flexible docking should be expected to be more accurate representations of reality.

There are two main challenges in docking method development: developing a search method that will be able to 'find' a near-correct docking orientation with reasonable

likelihood, and developing a scoring function/energy function that can discriminate correctly or near-correctly docked orientations from incorrectly docked ones.

The main procedure of rigid protein-protein docking includes the following stages: Firstly, obtain the coordinates for both two molecules. Based on the source of their component 3-D structures, docking algorithms are classified as "bound docking" or "unbound docking" algorithms. For the former, a protein complex is pulled apart and re-assembled. For the latter, the two 3-D structures of the docking partners could come from protein structures independently solved by experiment (X-ray, NMR) or come from two different complexes (that is, the docking partners are complexed with a molecule different from the one used for docking), or come from homology models where each docking partner model was built separately. In the second docking stage, the receptor (the molecule assumed to be stationary) and the ligand are treated as rigid bodies and the six rotational and translational degrees of freedom are fully explored with scoring functions that are tolerant to conformational changes. Finally in the refinement stage, a small number of structures obtained in the initial stage are refined and re-ranked using more detailed score functions that take into account conformational changes. Frequently, conformational searches involving side-chain rotamers and energy minimizations are performed in the refinement stage.

2.5.2 Search Algorithms:

To predict how the two molecules might fit together, docking programs usually assume that one molecule is kept stationary and undertake a six-dimensional search with three degrees of translational freedom (translation along x, y and z axes) and three degrees of rotational freedom (rotation around the x, y and z axes) for the other mobile molecule. In flexible protein-ligand docking the search process also involves

exploration of the torsion degree of freedom of the ligand. The number of possibilities for putting two molecules together grows exponentially with their sizes. The search for candidate solutions in a docking problem is addressed in two essentially different ways: either by applying a complete solution space search method, or through a gradual guided progression through solution space, in so called "constraint-based" methods. Constraint-based methods are widely used in protein-ligand docking programs and usually require that the binding site of the receptor molecule is known. They either only scan part of the solution space in a partially random and partially criteria-guided manner, or generate a fitting solution. The underlying idea in most constraint-based methods is that the binding site can be described by a series of points that are to be matched by the ligand with respect to either geometric or force-field complementarity or both. The search of ligand conformational space is constrained to the location of these points. Usually Monte Carlo/simulated annealing, molecular dynamics, evolutionary algorithms, and fragment-based method, are applied. Monte Carlo simulation is one of the most widely used statistical simulation methods. It is distinguished from other simulation methods by being. stochastic, in that it utilizes sequences of random numbers to perform the simulation of processes that can be described by probability density functions (pdfs). Simulated annealing is a generalisation of the Monte Carlo method and is suitable for optimizing problems of large scale. Molecular Dynamics is simulation of molecular motion by addressing the numerical solutions of Newton's equations of motion, and is therefore a special discipline of molecular modelling. Evolutionary algorithms are stochastic search methods based on ideas borrowed from genetics and natural selection and especially suitable for solving difficult optimizing

problems. In fragment-based methods, the ligand is divided into fragments. These fragments are docked to the receptor separately, and finally linked together. DOCK was one of the first constraint-based docking methods and is widely used (Kuntz ID *et al*, 1982; Kuntz ID, 1992; Meng EC *et al*, 1992; Shoichet BK, *et al.*, 1992). Fragment-based searching method was applied in DOCK. Initially DOCK treated the ligand as a rigid-bogy. In the later version of DOCK the ligand was treated as flexible via incremental construction of ligand in the binding area. DOCK is mainly used for protein-ligand docking. Sometimes DOCK is used for protein-protein docking.

Six-dimensional complete space searches (global) are not computationally tractable for protein-size molecules. Some level of simplification of the protein 3-D presentation and computationally efficient search method must be used in protein-protein docking programs. One of the most commonly used search methods in this area is the "geometric hashing" based matching algorithm focusing only on the relevant conformational space and is therefore computationally very fast (Fischer D, *et a*l, 1995). Another widely used method in docking algorithms employs Fourier correlation techniques that greatly reduce the complexity of the translational scan, for example three-dimensional grid based Fast Fourier Transform (FFT) (Katchalski-Katzir E., *et al.*, 1992). When dealing with very large complexes, geometric hashing algorithms are prone to a combinatorial increase in the number of features that need to be compared as the size of the molecule increases, and computing with single solution FFT can be very slow. Some protein-protein docking algorithms such as Hex apply spherical polar Fourier correlations to deal with this problem (Ritchie DW & Kemp GJL, 2000).

Program	Search Algorithm	Scoring Function	reference
Dock	Fragment-based	Grid-based energy function	Kuntz, *et al*, 1982; Meng, *et al*, 1992.
3D-Dock	FFT	Shape complementarity + electrostatic + residue level pair potentials	Aloy *et al*, 1998 Moont *et al*, 1999
DOT	FFT	Electrostatic + Van der Walls force	Mandell *et al*, 2001
GRAMM	FFT	Geometric Fit	Katchalski-Katzir E, et al, 1992; Vakser, 1995
ZDOCK	FFT	Pair-wise shape complementarity + desolvation + electrostatic	Chen R. Li L & Weng ZP, 2003;
Hex	Spherical polar Fourier	Shape complementarity	Ritchie & Kemp, 2000
BIGGER	Geometric Hashing (bit mapping)	Shape complementarity + electrostatic + desolvation + pair potential	Palma *et al*, 2000.
ClustPro	FFT	Desolvation + electrostatic energy	Comeau *et al*, 2004.
RosettaDock	Monte Carlo	Packing interaction, salvation, rotamer probability, hydrogen bond, electrostatic	Gray *et al*. 2003

Table 2.1 Established protein-protein docking programs, the search algorithm and scoring functions they employ, and their authors. These programs are mainly rigid-body protein-protein docking programs except *Dock* and *RosettaDock*. DOCK is mainly used for protein-ligand docking. Its initial versions were rigid-body docking. Incremental construction technique was applied in the later version to introduce ligand backbone flexibility. Both of their current versions are flexible docking. RosettaDock is flexible protein-protein docking program which is included in this table because it will be used as a comparison/validation with/of our predictions using rigid-body docking programs.

2.5.3 Scoring Functions:

The search stage of molecular docking creates a population of solutions. Then a scoring function is used to measure how good each solution is. Different methods apply different physico-chemical criteria to calculate various docking scores considering, for example geometric complementarity, contact area, inter/intra-molecular overlap, hydrogen bonds, electrostatic interactions, pair-wise amino acid contacts, solvation energy, active site residues, free energy, and other interaction properties (see below) (Halperin I, *et al*, 2002).

The same forces that contribute to driving separate proteins together to form complexes have a stabilizing effect in the complex structure. Docking scoring functions aim to take into account the strength of these forces while evaluating the plausibility of the docking solution. For example, the effects of electrostatics in protein-protein association are usually calculated by Poisson-Boltzmann equation, one of the most popular continuum models for describing electrostatic interactions between molecules in salty, aqueous media. Poisson-Boltzmann equation will be explained in more detail later in the chapter 2.6.

In addition, scoring functions often take into account other important properties of the contact surface between the molecules. For example, shape complementarity which calculates geometric complementarity and inter/intra-molecular overlap contributions in rigid-docking programs. Inter-molecular overlap is a balance to geometric complementarity. Intra-molecular overlap is calculated when the ligand or receptor backbone flexibility is taken into account. The binding interfaces of native protein-protein complexes do not necessarily have the largest extent of buried surface areas in comparison with artificial complexes generated with docking programs

(Norel R, *et al.* 1999). Native complexes also do not necessarily have the largest non-polar buried surface areas, or the largest number of hydrogen bonds, or the smallest number of unsatisfied buried polar groups. In solution, unsatisfied buried polar groups are likely to induce surface motions to eliminate largely such unfavourable energy contributions (Halperin I, *et al.*, 2002).

Other information, such as pair-wise amino acid or atom-atom contacts, and interface residue type, etc. may be included in the scoring, or help to reduce the number of allowed solutions.

Scoring function might be employed at the search stage to filter emerging solutions, in which case it can be called *integrated* function, or it may be used only after the search stage to order all the emerging solutions, in which case it can be termed as *edge* function (Halperin I, *et al.*, 2002). Integrated scoring functions are expected to have more impact on the docking results than edge functions.

Although some docking programs are often able to rank correct solution within the top two hundred, or even with the top ten solutions, in most cases the highest ranked structures are still false positive. There are no reliable methods currently available for discriminating the correct solution from false positive solutions (Norel R, *et al.*, 1999).

2.5.4 Assessment of different dock programs:

Most docking program developers test their methods on crystal structures from the Protein Data Bank. In the most favorable cases, the best prediction and the returned model of the complex structure is reasonably similar to the crystal structures. None of the methods achieve this on all test structures, especially when unbound molecular

conformations, which can differ noticeably from the bound conformations (either in side-chains or main-chains or both.), are used to make predictions.

The community-wide experiment Critical Assessment of Predicted Interactions (CAPRI) aims to assess the performance of procedures for predicting the mode of association of two proteins based on their three-dimensional structures (Janin J, 2002). CAPRI is a blind test of the ability of protein-protein docking programs. It starts whenever the experimental structures of the components are known and that of the complex is made available only at the time of evaluation. In four years, 17 crystal structure complexes prior to publication were subject to structure prediction by docking their two components (Janin J, 2005) as part of CAPRI. Docked models of these complexes were evaluated by comparing their geometry to the crystal structures, and by evaluating the quality of the prediction of the interaction regions and pair-wise residue contacts. The structural fit between predicted complexes and observed complexes can be evaluated by several parameters, for example backbone RMSD (root mean square deviation). In the evaluation of CAPRI, the receptor structures were superimposed first, and then the RMSD. of ligand molecule of the predicted versus the experimental complex is calculated. The RMSD calculation is based on backbone atoms (Ca, N, C and O atoms).

A pair of interface residues on different subunits was defined to be in contact if any of their atoms is within 5Å distance. Residue-residue contact performance of the docking predictions was measured by two parameters. One was the percentage of correct contacts defined as the number of correct residue-residue contacts in the predicted complex divided by the number of contacts in the experimentally determined complex. The other was the percentage of incorrect contacts defined as

the number of incorrect residue-residue contacts in predicted complex, divided by the total number of contacts in this complex. Another criterion to assess whether the two molecules interact using the native interface in the predicted complex is the fraction of correct interface residues. This is defined as the number of correct residues in the predicted interface divided by the total number of correct residues in the corresponding crystal structure.

Prediction was successful for 12 of the 17 complexes between all the programs. Most of the failures were due to large conformational changes that the programs (all rigid-body protein-protein docking) could not cope with.

2.6 Molecule Surface

Docking programs attempt to reproduce the interaction occurring at the protein surfaces upon complex formation. Therefore, an important question is how to mathematically represent a protein surface.

2.6.1 Molecular Surface Representations

A structural diagram of molecules can be easily obtained by line drawing or wire models. In CPK (space-filling) type models, the spherical representation of the atoms in a molecule along with the valence geometry at a particular atom draw one's attention to the surface of the atoms and the molecule instead of to the nature of the bonding and therefore is perhaps the simplest type of representation of molecular surface.

A major advance in the representation of protein surfaces termed as "solvent-accessible surface" was initiated by Lee and Richards (Lee and Richards, 1971). Solvent accessible surface is defined by ``rolling" a spherical water molecule probe

over the protein atoms' spherical surface (using the Van der Waals radii for individual atoms). In a way it is an expanded van der Waals surface generated by increasing each atom's van der Waals radius by the probe radius (figure 2.1). Later Richards further refines the surface definition by introducing the "contact surface" (highlighted in red in figure2.1) and "re-entrant surface" (highlighted in blue in figure2.1), and combining them together to form the "molecular surface" (Richmond and Richards, 1978; Richards, 1977). The contact surface is that part of the van der Waals surface of each atom which is accessible to a probe sphere of a given radius. The re-entrant surface is the inward-facing part of the probe sphere when it is simultaneously in contact with more than one atom (Connolly ML, 1996, www.netsci.org/Science/Compchem/feature14e.html).

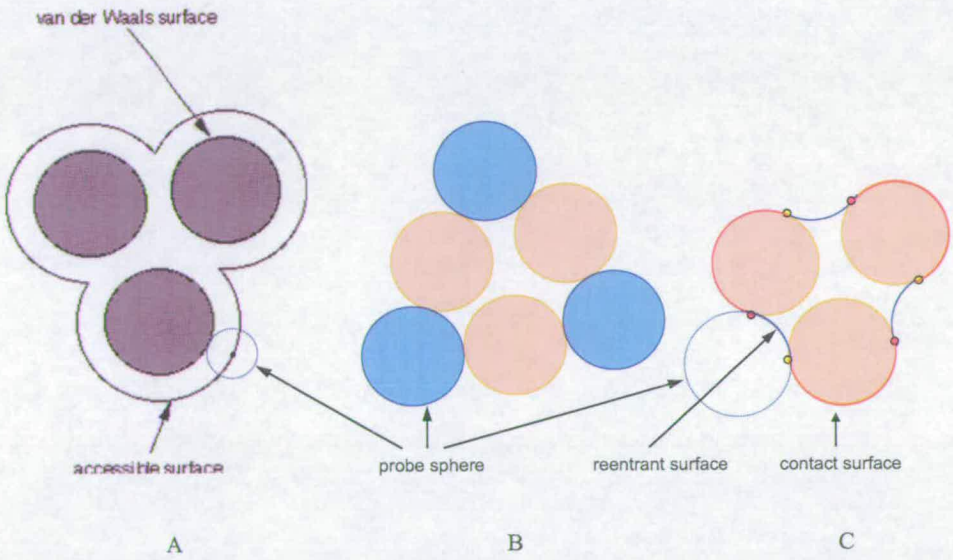


Figure 2.1 Representations of accessible surface (A) and molecule surface (C). Molecular surface is equal to "reentrant surface + contact surface". In the C part of this diagram, reentrant surface is highlighted in blue, and contact surface in red. This picture was modified from www.netsci.org/Science/Compchem/feature14e.html.

While Lee and Richards's solvent-accessible surface presentation is now widely used, it is not suitable for docking because it add a probe radius to each atom's van der Waals radius of the receptor/ligand molecule and results in surface crevice into which receptor/ligand atom can intrude (figure 2.1). Docking programs are programmed to model molecular surfaces according to the Connolly program (Connolly, 1983a and 1983b), which calculates Richmond and Richards' "molecular surface" of a molecule given the coordinates of its atoms, specified van der Waals radii for different atoms, and the probe radius. Connolly surfaces with probe radius 1.4 Å are the most commonly used solvent-accessible surfaces in the field. One of the advantages of the molecular surface over the accessible surface is its ability to visualize the shape complementarity at interfaces :

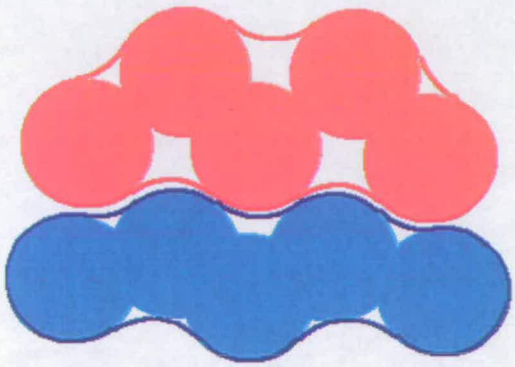


Figure 2.2 Shape complementarity at interfaces. This picture is taken from http://www.netsci.org/Science/Compchem/feature14e.html.

Other surface descriptions based on B-splines and Gregory patches have been developed (Blinn, JF, 1982; Colloc'h N and Mornon JP, 1990). Each of these definitions represents component atoms as hard spheres. The hard sphere

representation is an abstraction of the actual atomic structure since the electron density of each atom has no sharp boundary.

2.6.2 Electrostatic Potential Surface Property

The electrostatic properties of biological macromolecules in aqueous solution are relevant to a large variety of biochemical processes. In protein-protein interactions, electrostatic complementarity is thought to be important for determining specificity, even in cases where it does not make a large contribution to the affinity. Electrostatic potential, caused by charged side-chains and bound ions, correlates with dipole moment, electro-negativity, and partial charge. Visualizing the electrostatic potential on a molecular surface is a fast and convenient way to compare molecules in this respect.

Electronegativity is defined as "the power of an atom in a molecule to attract electrons to itself" (Pauling L, 1993). Approximately, the closer an atom is to fluorine in the periodic table, the greater is its electronegativity, and the greater the electronegativity difference between two atoms in a chemical bond, the more polar is the bond. The charge distribution over a molecular surface can be partitioned into atom-centred partial charges. Partial charges are affected by electronegativity; the most electronegative atoms are most negative, the others are less negative or more positive. The larger the difference in partial charges in a molecule, the more polar it is. The standard modern way to calculate partial charges is to perform a quantum chemical calculation. A least squares fit procedure is then used to produce a set of partial charges producing potential values most consistent with the quantum calculations (Cieplak P, et al. 1995). The methods that have been used to compute

electrostatics in biological systems may be broadly classified into those which explicitly simulate all molecules of the system, including salts and solvent (which are by far the more demanding), and those which simulate the solvent and salts through a continuum model. Among the latter, the Poisson-Boltzmann equation (PBE) has been widely and successfully used. The Poisson-Boltzmann equation was first put forward nighty years ago (Gouy M, 1910; Chapman DL, 1913), a combination of the Poisson equation and the Debye-Huckel theory. Refined theoretical and numerical tools were developed later to apply the PBE to biomolecular systems by different groups (Gilson MK, et al. 1987; Sharp KA and Honig B, 1990; Madura, et al. 1995). Most contemporary programs for calculating electrostatic surface potential, including GRASP (Nicholls A., *et al.*, 1991), use a continuum theory to model the interaction of solute molecule with the solvent. The electrostatic potential inside and outside the molecule can be obtained by solving the Poisson-Boltzmann equation (EQ 2.1).

$$\nabla \cdot [\varepsilon(\vec{r})\nabla\varphi(\vec{r})] + 4\pi\sum_{s} c_s(\vec{r})q_s \exp\left[-\frac{q_s\varphi(\vec{r})}{k_B T}\right] + 4\pi\rho(\vec{r}) = 0$$

EQ 2.1

where $\varepsilon(\vec{r})$ = dielectric constant

$\varphi(\vec{r})$ = electrostatic potential

$c_s(\vec{r})$ = concentration

q_s = charge of ionic species

k_B = Boltzmann's constant

T = temperature

$\rho(\vec{r})$ = charge density of the solute, this can be separated into two components, external and internal.

2.6.3 Hydrophobicity Potential Surface Property

As hydrophobic effect plays an important role in protein-protein interaction in an aqueous environment, it is important to devise a measurement for it. The characterisation of protein surface as being hydrophobic or hydrophilic has generally been done on the basis of the underlying atoms. Most commonly hydrophobicity is assigned by residue type. A hydrophobic residue contributes hydrophobic surface, a hydrophilic residue contributes hydrophilic surface. A way forward would be to calculate a potential (as is common in electrostatics).

The overall hydrophobicity (measured as lipophilicity in some programs such as SYBYL (Tripos Inc.)) of a molecule can be measured by its partition coefficient (logP) in polar/apolar heterogeneous reference systems. An example of a lipophilicity potential is used by the program SYBYL (SYBYL manual):

$$\log P = \mathrm{Sum}(f_i);$$

where atomic partial values can be regarded as fragmental increments, f_i, to the total lipophilicity given by logP.

A lipophilicity potential parameter (LP) is sometimes used to describe how lipophilicity is distributed over the different parts of a molecule (to produce lipophilicity maps and distinguish hydrophilic and lipophilic regions of a molecule). The average LP can be considered as an estimate of the logP of a molecule in octanol/water. LP is defined by considering a molecule S surrounded by non-polar or low polarity organic solvent molecule L. The arrangement of the solvent molecules L around S varies from a random distribution at far distances to an ordered distribution at short distances.

$$LP_{AC} = \mathrm{Sum}(fi \times g(d_i));$$

$$LP_{HM} = Sum(f_i \times g(d_i))/Sum(g(d_i);$$

with d_i =distance of a certain point in space from atom i.

and $g(d_i)=1/1+d_i$

where LP_{AC} is applicable to small molecules and LP_{HM} to big molecules.

2.7 Protein sequence comparison

Protein sequence comparison methods are an essential pre-requisite to protein structure prediction methods. Proteins are products of evolution. Their sequences, via mutation and conservation, are under evolutionary selection pressure to maintain structure and (in many cases) also function can tell us which particular residue could be important for structure and/or function. Therefore, comparing different, but probably evolutionarily related, sequences can yield important information for the functions of proteins.

A variety of comparison algorithms and scoring parameters can be used to evaluate protein (or DNA) sequence similarity. Overall, these comparison algorithms can be roughly categorized into two groups: pair-wise comparison algorithms, and multiple comparison algorithms including sequence profile methods.

2.7.1 Pair-wise Sequence Comparison

The similarity between protein sequences can be displayed by a similarity matrix. An example for a simple similarity matrix is a 2-D-matrix with the two sequences to be compared along the vertical and horizontal axes. Pair-wise comparison algorithms try to find a single path through the matrix aligning the largest number of identical and biologically similar residues without opening too many and too long gaps. The most popular algorithm to do this is the global dynamic programming algorithm

(Needleman SB and Wunsch CD, 1970). The first step in the global alignment dynamic programming approach is to create a matrix with M columns and N rows where M and N correspond to the numbers of residues in the sequences to be aligned (figure 2.3). The two sequences are compared along the vertical and horizontal axes, respectively, with the unit of the axes being sequence residues. Individual cells in the matrix store a score representing the similarity between the two residues. In the second step, the matrix fill step, one possible (inefficient) solution finds the maximum global alignment score by starting in the upper left hand corner of the matrix and following the maximal score for each position in the matrix. After the matrix fill step, the program traces back from the lower right corner to the upper left corner and identifies the highest scoring path through the matrix, penalising for insertions/deletions at the same time. During the dynamic programming process, substitution tables are used to evaluate similarities between different residues in aligned pairs, and gap penalties are applied to penalise insertion/deletions; This prevents excessive insertion of gaps in the alignment. The scores in the cells of the matrix usually came from substitution tables. The most widely used substitution tables are the Dayhoff matrix (Dayhoff MO, *et al*, 1978), which is an example of the so-called PAM (Percentage of point accepted mutation per 108 years) matrices, and the series of BLOSUM (BLOcks SUbstitution Matrix) matrices (Henikoff S and Henikoff JG, 1992).

	M	D	S	E	V	D	--	L	
M	5	-3	-1	-2	-2	-3	--	2	*1*
E	-2	2	0	5	-3	2	--	-3	*2*
T	-1	1	1	0	-2	1	--	-2	*3*
G	-3	-1	0	-2	0	-1	--	-4	*4*
E	-2	2	0	5	-3	2	--	-3	*5*
D	-3	6	0	2	3	6	--	-4	*6*
--	--	--	--	--	--	--	--	--	--
L	2	-4	-2	-3	-4	-4	--	4	*N*
	1	*2*	*3*	*4*	*5*	*6*	--	*M*	

Figure 2.3 An example of similarity matrix M×N. M is the number of residues in the sequence along the horizontal axis of the matrix, N is the number of residues in the sequence along vertical axis of the matrix. The number in the each cell is the similarity score of the two corresponding residues.

PAM matrices are based on the frequency of observed residue pairs in protein pairs that are similar enough to be aligned reliably. The substitution scores applied in pair-wise sequence comparison are thus derived from a mutation probability matrix where each element gives the probability of the amino acid in column X mutating to the amino acid in row Y after a particular evolutionary time. Evolutionary time is expressed in PAM units, which (unlikely percentage sequence identity/divergence) take into consideration the possibility of "back-mutations", namely the residue undergoes an amino acid change during evolution that is reverted later, resulting in a seemingly conserved residue. A PAM matrix is specific for the particular evolutionary distance of its underlying set of sequences, but may be used to generate matrices for greater evolutionary distances by multiplying it repeatedly by itself (Or, conversely, for smaller distance through division). The version that Dayhoff published was extrapolated to be a PAM250 matrix. It is possible to go above

PAM100 because if one residue changes several times, each change is counted. PAM matrices for larger evolutionary distance, and/or those derived from quite divergent sequences, are more sensitive for detecting homologous sequences than PAM matrices derived from closely related sequences because they give less priority to identical amino acid matches and more to conservative substitution. However, at large evolutionary distances the information present in the matrix is essentially degenerated. It is rare that a PAM matrix would be used for an evolutionary distance any greater than 250 PAM units.

Dayhoff's methodology of comparing closely related species turned out not to work very well for aligning evolutionarily divergent sequences. Sequence changes over long evolutionary time scales are not well approximated by compounding small changes that occur over short time scales. The BLOSUM series of matrices aims to rectify this problem. Henikoff and Henikoff constructed these matrices using multiple alignments of evolutionarily divergent proteins. Unlike Dayhoff's PAM matrices developed from global multiple alignments (Note: not all PAM matrices are based on global alignments), the BLOSUM matrices are based on local multiple alignments of more distantly related sequences. For example, BLOSUM 62, the default matrix in most implementations of BLAST, is a matrix calculated from comparisons of sequences with no less than 62% pair-wise sequence identity. Different BLOSUM matrices are not extrapolated from existing BLOSUM matrices, but are always based on local multiple alignments. For example, the BLOSUM 80 matrix was derived from a set of sequences having 80% sequence identity. Higher identity BLOSUM matrices are more suitable for aligning two closely related sequences while lower identity matrices should be used for more divergent sequences.

The BLOSUM62 matrix has been proven to be a good initial choice for a wide range of sequence diversity and problems; this matrix is selected by default in many widely used search and/or alignment applications.

Gap penalties are another important consideration in the scoring of pair-wise alignments. There are various ways of incorporating gap penalties. The most common gap penalty schemes are length-based; a fixed penalty is charged for opening a gap and that penalty increases with the length of the gap. Other schemes are length-independent (Orengo CA, *et al*, 2003).

Large proteins are often only evolutionarily conserved in local regions. For example the homology between two multi-domain proteins may only extend over a single domain. Therefore most alignment methods adopt a local dynamic programming strategy and seek local regions of similarity between protein sequences. Instead of looking at the each sequence in its entirety, the widely used Smith-Waterman algorithm (Smith TF and Waterman MS, 1981) compares segments of all possible lengths and chooses whichever maximise the similarity measure. This algorithm is implemented in the popular program SSEARCH (Smith TF and Waterman MS, 1981(b)) and MPsrch (Sturrock S, Collins J, 1993) which can be used to detect homologous domains among multi-domain proteins.

Generally, dynamic programming is considered very reliable for comparing sequences and finding their optimal alignment. However, global dynamic programming is computationally very slow and though local dynamic programming can speed up the search process, it is still computationally very expensive for scanning large sequence databases. Therefore, alternate strategies have been developed that implement heuristics to derive results more rapidly.

A heuristic algorithm delivers an approximate solution to a given problem. Sometimes it was not possible to formally prove that this solution actually solves the problem, but heuristic methods are commonly used because they generally are much faster than exact algorithms. Two rapid heuristic algorithms are widely used for searching protein sequence databases, FASTA (Pearson W.R. and Lipman D.J., 1988) and BLASTP (Altschul *et al.* 1990). FASTA employs a heuristic short-cut for selecting limited regions in a limited set of database sequences and then performs a Smith-Waterman alignment within these regions. BLASTP uses another heuristic that enables it to skip most of the database. It then searches for high-scoring pairs of short segments and connects them with a gapped alignment.

The suite of BLAST programs consists of nucleotide BLAST (BLASTN), protein BLAST (BLASTP), translated BLAST (BLASTX for nucleic acid query protein database; TBLASTN for protein query-translated database; TBLASTX for nucleic acid query translated nucleic acid database) (Altschul SF, et al, 1990). The similarity between two sequences is measured with several scores depending on the BLAST alignment: E-value, Bit score and P-value. E-value (Expectancy Value) is equal to P-value multiplying with the size of the database being searched. The E-value is related to the probability that the observed degree of similarity could have arisen by chance: it is an estimation of the number of the sequences that would be expected to match as well or better than the one being considered. The higher the E-value, the lower the similarity between the two sequences.

2.7.2 Multiple Sequence Comparison

Multiple sequence alignments can be used to study groups of closely and distantly related proteins, by identifying patterns of conservation and variation. For example,

if some residue positions that are much more conserved across a superfamily than others, they may well be associated with functional sites or important surface patches. Dynamic programming algorithms are readily extended to multiple sequences. However, the computational time to run dynamic programming for more than three sequences is prohibitively expensive. Although there are still several programs, for example MSA (Lipman DJ *et al*, 1989; Gupta SK *et al*, 1995) and SAGA (Notredame C and Higgins DG, 1996), that use dynamic programming to get guaranteed best alignments for up to eight sequences with maximum length of 100 residues, most multiple alignment methods today are based on different approaches to build up the alignments.

2.7.2.1 Progressive Pair-wise Alignment Approaches

The progressive pair-wise alignment algorithm (Feng DF and Doolittle RF, 1987) is applied iteratively to generate a multiple alignment of the given proteins and to construct an evolutionary tree depicting their relationship. The closest sequences according to this evolutionary tree are aligned first. Then in some approaches, for example CLUSTAL W (Thompson JD, *et al*, 1994), other sequences are adding to this pair-wise alignment in order of the tree whereas some approaches, for example T-Coffee (Notredame C, *et al*, 2000), first align all closely related sequences pair-wise and derive a consensus sequence of residues aligned at each position and then align the consensus sequences.

CLUSTAL W is still one of the most commonly used programs and uses a progressive pair-wise alignment approach. Its progressive strategy is to derive an initial, approximate, phylogenetic tree between the sequences to work as the "guide tree". This guide tree is calculated from the pair-wise distance matrix of these

sequences using the Neighbour-Joining method (Saitou N and Nei M, 1987). The progressive alignment is accomplished by gradually building up the alignment by selecting a starting pair and aligning them, and then each subsequent sequence is aligned to the previous alignment, following the order suggested by the guide tree. Automatically variable scoring matrices are used for each alignment based on the expected evolutionary distance. Gap penalties are also automatically variable and depend on the scoring matrix used, sequence similarity, context (hydrophilicity, presence of gaps in other sequences), sequence lengths, and difference in sequence lengths. CLUSTAL W works reasonably well in a wide variety of cases. However, once an error is made in the first alignment, it cannot be rectified later when the rest of the sequences are added in (Notredame C. *et al.*, 2000).

T-Coffee is a more advanced method designed to avoid this pitfall. The program first pre-processes a data set of all pair-wise alignments between the sequences. In this way it obtains a library of alignment information that could guide the progressive alignment. Following this the multiple sequence alignment is generated, considering both the sequence to be aligned next and how all of the sequences align to each other. The alignment is derived using heterogeneous sources, such as a mixture of alignment programs and/or structure superposition which is an additional innovation.

Some protein sequences share a similar region but are otherwise completely different. Several local multiple alignment algorithms, for example MACAW (Schuler GD, *et al*, 1991) and Gibbs (Lawrence CF, *et al*, 1993), have been developed to deal specially with these cases. MACAW, which uses the BLAST algorithm, tries to find high scoring segment pairs (HSPs) for each possible pair of sequences and then assembles overlapping HSPs into blocks. The Gibbs algorithm iteratively derives a

profile with stretches of n residues selected from sequences and used this profile in searches against one of the other sequences. The result of every search cycle is used to weight the selections of the stretches in the next cycle.

2.7.2.2 Probabilistic Approaches (Statistical Profiles and profile Hidden Markov Models)

Probabilistic approaches to multiple sequence alignment use additional information for creating multiple alignments compared with the other methods. This information is derived from the existing multiple sequence alignment of similar sequences in each step. Typically this information is a statistical profile, basically a record of probability of finding a given amino acid at a given alignment position, and such profiles are used for generating a new alignment (Orengo CA, *et al*, 2003, chapter 5). Statistical profiles are like scoring matrices which also reflect probabilities of one amino acid changing to another but profiles specify these probabilities for each amino acid position of alignment separately. Different profiles need to be created for different families of proteins. Statistical profiles may also take other factors into account, for example variable gap opening and extension penalties. A typical example of the use of profiles for sequence alignment is the program PSI-BLAST (Altschul SF, *et al*, 1997), which creates alignments and profiles to detect more distant members of a given family in the selected protein sequence database.

Even more advanced methods are Hidden Markov Model (HMM) based methods which can be useful for identifying very distant relatives. A HMM is a dynamic kind of statistical profile. It uses a sequence model to store the probabilities of amino acid substitutions and frequencies. A HMM is similar to a statistical profile in that each column in a statistical profile can be seen as a match state and the values in the

column as emission probabilities for each of the 20 possible amino acids. The main difference between a HMM and a profile is that the profile model requires that the transitions from a match state to an insert state or a delete state have the same probability (Orengo CA, *et al*, 2003, chapter 5). Generally gap penalty values are also coded into HMM model. This characteristic of HMM, together with its alignment time depending linearly, as opposed to exponentially, on the number of sequence, make HMM methods particularly useful for aligning protein families with many members. The most widely used profile HMM based programs are HMMer (Eddsy SR, 1998) and SAM ((Hughey R & Krogh A, 1996; Karplus K et al, 1998)). Generally, pair-wise comparison algorithms are useful for detecting close homologues, for example BLAST and FASTA. For detecting more distant homologous, pair-wise alignment is no longer reliable as the signals in the 2-D path matrix is often very weak and most algorithms have difficulty in identifying the optimal path. Under these cases, more powerful sequence alignment can be performed by profile and HMM based alignment methods. Profile HMM based alignment methods are among the most powerful methods for protein homology detection.

2.8 Protein Structure Prediction

The 3-D structure of a protein can yield important information about its function. However there exits a huge gap between the number of proteins with experimentally determined structures and the number of known protein sequences in the GenBank (Benson DA, *et al*, 2004) sequence database. There were approximately 67,050,181 sequences stored in GenBank till February 2006. At the same time, there are only

36932 structures in the PDB (Berman HM, *et al*, 2000) till June 2006. The main reason for the discrepancy is that, by comparison with sequence determining methods, experimental methods for determining protein structure are expensive and time-consuming. For some proteins, for example many membrane proteins, experimental difficulties can be so substantial that their structures remain unresolved. Therefore a great number of researchers have been interested in finding a method for predicting the native structure of a protein given just its sequence, from the time the first experimental protein structure was published in 1957 (Kendrew JC et al, 1958).

2.8.1 Protein Structure Prediction Approaches

Protein tertiary structure prediction methods are usually divided into several categories: *Ab initio* folding methods that use only the information in the target sequence itself; fold recognition (or threading) methods that are based on the observation that there is a limited number of naturally occurring protein folds; and comparative modelling that approximate the 3-D structure of a target protein sequence based primarily on its alignment to one or more homologous proteins with know structures (templates).

Ab initio folding methods can be grouped to distinguish two different types of approaches: One is the "knowledge-based" method group that tries to extract rules of protein folding by observing known protein structures and then apply these rules to predict structures of proteins for which experimental structure data is not available. This group of methods should become more powerful as more and more experimental determined structures become available. The second group of methods tries to simulate the protein folding process based on the assumption that the native

fold of a protein can be found by finding the conformation of the protein with the lowest energy, as defined by a suitable potential energy function. There are two key challenges for approaches of this type, how to define the energy function and how to devise an algorithm capable of finding the global minimum of this function. Most energy functions take into accounts at least hydrogen bonds and van der Walls forces. Searching for the correct structure in the set of all the possible conformations is computationally expensive, and prohibitive for large molecules. Until now, most simulation techniques are primarily useful for short peptides and small protein molecules. However these techniques are potentially useful for predicting loop conformations which is often not accomplished satisfactorily by the template-based methods described below.

Protein fold recognition (or threading) is based on the observation that there is a limited number of naturally occurring proteins. Nonetheless there was about a 70% chance that a newly characterized protein which had no obvious common ancestry to proteins with a known fold will be able to find a suitable template structure to build a 3-D model (Orengo, Jones & Thornton, 2003).

Comparative protein structure modelling, by far the most reliable technique for predicting protein tertiary structure, allows us to build a three-dimensional (3-D) model for a protein of known amino acid sequence, but unknown structure, using another protein of known sequence and structure as a template. Comparative modelling is also known as 'homology modelling', implying that models are always generated from homologous proteins.

The general comparative modelling procedure consists of four steps (Marti-Renom, M.A., Yerkovich, B. and Sali A., 2002):

(1) Identification of known structures that are related to the target sequence to serve as templates. This step is facilitated by numerous protein sequence and structure databases, usually the PDB, and sequence similarity searching software available on the web, for example BLAST and FASTA; distant homologues may be identified by PSI-BLAST (Altschul SF, *et al*, 1997). Once a list of all related protein structures is obtained, it is advisable to select which templates are most appropriate for the specific modelling problem. Usually, higher overall sequence similarity between the target and template sequences will yield a better model. Several other factors that should be taken into account are: (a) Subfamily: if the list of sequences, (including the target sequence), can be divided into subfamilies, the template with the sequence from the sub-family that is closest to the target sequence should be selected. (b) 'Environment'. The term environment is used here to consider factors that determine protein structure, beside its sequence (e.g., solvent, pH, ligands, and quaternary interactions). The template protein's environment should be compared to the required target structure environment. (c) The quality of the experimental template structure.

(2) The second step, which is also the most important step in comparative modelling, is to align the target sequence with the selected template(s). There are a great variety of protein sequence alignment methods, many of which are based on dynamic programming techniques. Multiple alignments are generally more reliable than pair-wise alignments. Two frequently used programs for multiple sequence alignment are CLUSTAL and T-Coffee; both are available as web servers. When the sequence identity between the target and template sequences is high (>70%), the alignment is generally highly accurate though it dependents totally on other things most importantly numbers and variety of sequences. However, at lower level of sequence

identity (<40%) and with the increasing of numbers of insertions and deletions, obtaining the correct alignment becomes very difficult. In such difficult cases, profile Hidden Markov Model (HMM) might help to generate an acceptable alignment. HMMs are domain family models that can be used to identify very distant relatives. Some secondary databases represent full domain alignments as HMMs. For example, PFAM (Bateman A, *et al*, 2004) offers 'seed alignment' for each family which contains representative members of the family, and a full alignment of all members of the family. A HMM is constructed from the seed alignment using the HMMer2 software and used to detect other members in this family and build the full alignment. Superfamily (Gough J, *et al*, 2001) contains a library of HMMs that represent essentially all protein domains of known structure using the SAM HMM package. These HMMs can be used to build multiple alignments including additional sequences of distant homologues.

For modelling, the best alignment is the one which would be achieved if one had the structures of the two proteins, performed a structural alignment, and then derived a sequence alignment from it. In difficult alignment cases, it is often beneficial to rely on multiple structures and sequence information. Misalignments can be minimised by using a large number of homologous sequences, including sequences without known structure, to construct a multiple alignment. In addition one can make the best use of all known structure information by hand-correcting the automatic sequence alignment. Corrections can be made so that Indels (insertions and deletions) occur in loop regions rather than within secondary structural elements (α-helices and β-strands). If there is more than one template structure, one would first generate a structural alignment of these structures, then extract the multiple sequence alignment

from this structural alignment, and finally align the target sequence to this multiple sequence alignment.

(3) Model building: Once an initial target-template alignment has been built, a variety of methods can be used to produce a detailed 3-D model for the target protein. During the model building procedure, the first thing is to assemble a model from a small number of rigid bodies obtained from the aligned protein structures (Blundell TL, *et al.* 1987; Greer J, 1990). The boundaries of structurally conserved regions (SCRs), the structurally variable regions (SVRs) (which are usually the loop region), and side-chains that decorate the backbone are dissected. If only one template structure is available for a target sequence, the main-chain atoms in the structural core are assumed to be structurally conserved and are simply copied from this template structure to the model. When several templates are available, modelling methods superimpose the coordinates of all templates (Sutcliffe MJ, *et al.*, 1987). Then all SCR main-chain atoms in the target model are obtained by copying the coordinates of the template whose sequence is closest to the target. SVRs typically vary much more in length and amino acid composition, and structure. Also, even loops with exactly the same length and the same amino acid sequence could have very different conformations. There are three major approaches to predict SVR conformation (Orengo, Jones and Thornton, 2003). In the 1980s, it was popular to build the loop region conformation by hand aided by molecular graphics programs, and then refine them by energy minimization. Knowledge-based methods search structure databases and look for segments in known protein structures that fit onto the anchor residues with the SCRs (Jones TH & Thirup S, 1986; Cothia C and Lesk AM, 1987). The third approach is to undertake a conformational search using

techniques similar to those adopted in *ab initio* folding (Moult J & James MNG, 1986; Fine RM, et al., 1986; Bruccoleri RE & Karplus M, 1987). Some modelling software offers a combination of knowledge-based and conformational search approaches (Chothia C *et al.*, 1986; Martin ACR, 1999; Van Vlijmen HWT & Karplus M, 1987).

Generally approaches to predict SCR and SVR conformations only deal with backbone atoms. Side-chain atom conformations can be built by various protocols (Vasquez M, 1996). The simplest "maximum overlap" protocol is to inherit the template side-chain atoms' torsion angles to the identical residues of the target model and to then build the other residues in a single standard conformation. The "minimum perturbation" protocol preserves the backbone Phi (Φ) /Psi (Ψ) angles and the equivalent side chain chi (χ) angles and then rotates the side chain atoms to change their chi angles to relieve clashes (Shih HH, *et al*, 1985). In the "coupled perturbation" protocol side chain atoms are treated similarly as in the minimum perturbation protocol but the side-chain torsion angles of structurally adjacent residues are also rotated (Snow M & Amzel LM, 1986).

In building side chain conformation, there are two most important effects to consider, the coupling between main-chains and side-chains and the discontinuous nature of the distributions of side-chain dihehral angles (http://www.salilab.org/~andras/watanabe/node10.html). There are significant correlations between side-chain dihedral angle probabilities and backbone angle values (Dunbrack RL & Karplus M, 1993). SCWRL (Bower MJ, *et al*, 1997) is one of the most effective and accurate programs for adding side-chains to a protein backbone. It is based on a backbone-dependent rotamer library. The library provides

lists of chi1-chi2 pairs for residues at given Φ /Ψ values, and explores these pairs to try to minimize side-chain-to-backbone clashes and side-chain-to-side-chain clashes. The comparative modelling software we used, MODELLER (Sali & Blundell, 1993), adopts similar techniques to those described above to build models. MODELLER models the structural conserved and varied regions and applies molecular dynamics for refinement in a single step. The 3-D models produced by MODELLER also have to satisfy additional spatial restraints, given as a 'probability density function'. In principle, the restraints can be derived from a number of different sources, including homologous protein structures, low-resolution NMR experiments, rules of secondary structure packing, etc. The optimization is carried out by the "variable target function" procedure employing methods of conjugate gradient and molecular dynamic with simulated annealing

(4) The final step of comparative modelling is model evaluation. The quality of a model determines whether the information extracted from it will be reliable. There are two main sources from which errors can arise. One of the main sources is that the modelling method may fail to find the optimal conformation during the conformation search stage. The other one is that the scoring function may fail to identify the optimal conformation. MODELLER automatically gives a score relating to the perceived quality of the model structure. This score is named MODELLER's "objective function" and is reported in the model PDB file. Users of MODELLER usually generate 5-10 models for each target sequence and select the model with the lowest objective function.

A model's quality can be evaluated in a hierarchical manner (Sánchez R and Sali A, 1998). The basic assessment is whether the model has the correct fold. A model with

the correct fold will usually overlap structurally with the actual structure over at least 30%. The overlap percentage will be high if the correct template was chosen and if this template was aligned at least approximately correctly with the target sequence. A popular evaluation method of the overall accuracy of a model is based on the geometrical similarity between the model and experimental structures. Root mean square deviation (RMSD) is a conventional measure of geometrical similarity. The overall model accuracy can be estimated very approximately by the sequence percentage identity between template and target sequence.

The stereochemical quality of a model can be investigated by different methods, for example PROCHECK (Laskowski RA, *et al*, 1993), WHATCHECK (Hooft RWW, *et al*, 1996b), ERRAT (Colovos C, Yeates TO, 1993), PROVE (Pontius J, *et al*, 1996) etc, evaluate the stereochemical quality of the molecular structures. The features checked by these programs include main-chain bond lengths and bond angles, dihedral angles, chirality, and hydrogen bonds (PROCHECK); pair-wise non-covalently bonded interactions (ERRAT); contact dots used in a kinemage, packing information such as van der Walls interactions, hydrogen bonds, atomic bumping, deviations of the atomic volumes from the standard values (PROVE), etc. Some other programs test the plausibility of 3-D models generally, for example VERIFY_3D(Luthy R, *et al*, 1992). VERIFY_3D calculates the statistical preferences (called 3D-1D scores) of each of the 20 amino acids for the environment of each residue position in the model, such as what area of the residue is buried, what fraction of side-chain area is covered by polar atoms (O and N); and the local secondary structure;

2.8.2 Assessment of Different Comparative Modelling Programs

The Critical Assessment of Structure Prediction (CASP) (Moult J et al, 2003) and the Critical Assessment of Fully Automated Structure Prediction (CAFASP) experiments provide a way of measuring the success of many protein structure prediction groups in a quantitative way on a predefined set of structures. CASP provides participants with the amino acid sequences of proteins whose structures are close to being determined experimentally by other researchers. The participants then predict and submit model structures for these target proteins using computer programs and often manual refinement. Finally when these become available, the prediction models are compared with structures from experimental studies. The evaluation of the model is usually divided into three different categories at CASP: comparative modelling, fold recognition and *ab initio* methods.

MODELLER (Sali A and Blundell TL, 1993) was the most popular comparative modelling package in the last two CASP experiments (Venclovas C *et al*, 2003; Kryshtafovych A *et al*, 2005). SWISS-MODEL (Schwede T, *et al*, 2004) and 3D-JIGSAW (Bates PA, *et al*, 2001) are also widely used. In the cases where the sequence percentage identity between target and template is very high, the models could be closer to the native structure than its template structure (Tramontano A, *et al*. 2001; Tramontano A and Morea V, 2003). Wallner and Elofsson (Wallner B and Elofsson A, 2005) undertook a large benchmark assessment of different comparative modelling programs including MODELLER, SegMod/ENCAD (Levitt M, 1992), SWISS-MODEL, 3D-JIGSAW, nest (Petrey D, *et al*, 2003), and Builder (Koehl P and Delarue M, 1994). Geometrical criteria and structural similarity to the experimental structure were used to evaluate these programs. In their test, MODELLER, nest and SegMod/ENCAD perform better than other methods

although the side-chain atoms these programs built were not as good as those by SCWRL (Canutescu AA, *et al*, 2003).

2.9 Background to Interaction between CDKs and cyclins

Cellular signalling pathways in eukaryotes are indispensable to the process of tissue growth, cell differentiation, and rapid response to environmental changes. Many signals are transmitted by phosphorylation and dephosphorylation of proteins through the mediation of protein kinase domains. Protein kinases of eukaryotes are subdivided into an exclusively eukaryotic protein kinase group (abbreviated as 'ePK'), a histidine protein kinase group, and an atypical protein kinase group (abbreviated as 'aPK').

Kinase classification is based on sequence comparison of their catalytic domains, aided by sequence and structure information outside the catalytic domains and known biological function. Hanks and Hunter classified human kinases into five broad groups, 44 families, and 51 subfamilies (Hanks S.K. and Hunter T., 1995). Manning's Group extended this classification to 9 groups, 134 families and 196 subfamilies by identifying 478 human ePK and 40 aPK genes (Manning G. etc. 2002).

The catalytic domain of protein kinases, which is the part adopting the so-called "protein kinase fold", is extremely well conserved among serine/threonine and tyrosine kinases. The protein kinase fold is structurally formed by two lobes: a small N-terminal lobe with a central anti-parallel β-sheet, and a relatively large C-terminal lobe which consists mainly of α-helices. In between the two lobes is the linker part, which contributes to a deep ATP-binding catalytic cleft and includes a highly

conserved phosphate binding loop, the activation loop (Huse M. and Kuriyan J., 2002).

Among the estimated 1,000 to 2,000 human protein kinases, a family of kinases activated by members of the family of cyclins, the cyclin-dependent kinases (CDKs), have been extensively studied because of their essential role in the regulation of cell proliferation, neuronal and thymus functions and transcription (Morgan D, 1997; Meijer L, *et al*, 2000).

CDKs are Serine/threonine protein kinases that phosphorylate the OH group of serine or threonine. They catalyze the reaction: ATP + a protein = ADP + a phosphor-protein. While serine/threonine kinases all phosphorylate serine or threonine residues in their substrates, they select specific residues to phosphorylate on the basis of residues that flank the phosphor-acceptor site (http://en.wikipedia.org).

The characteristic originally defining the CDK-family is their requirement of cyclin binding for activity. Both the CDK and cyclin super-families contain many paralogous members in eukaryotes. In the cells of higher eukaryotes, a large number of different CDK-cyclin complexes with various substrate specificities form at various cellular locations and various time points of the cell cycle to thoroughly and finely tune the cell cycle. The substrates of CDK-cyclin complexes include transcription factors, nuclear matrix, nuclear membrane proteins, cyto-skeleton proteins, and other cell cycle proteins.

Beside their substrates, CDK-cyclin complexes are also able to interact with inhibitors and other cyclins. For example, cyclin F could regulate the nuclear localization of cyclin B1 through interaction and form a complex consisting of cdc2-cyclin B1-cyclin F (Kong M, *et al*, 2000); A different inhibitory association of the

Cdc2-cyclin B1 complex with the p53-regulated protein Gadd45 has also been reported (Zhan Q, *et al*, 1999).

2.9.1 CDK/cyclin structure-function relationship

CDKs are Ser/Thr kinases (about 300 amino acids in length, molecular weight 33-40 kDa) which display the typical structural protein kinase features. Like all protein kinases, the catalytic core of human CDK2, consists of two lobes. The small N-terminal lobe consists of about 85 residues with mainly β-sheet structure and a single large helix (α1) on which the "PSTAIRE" motif is located. The larger C-terminal lobe contains six α-helices and a small β-ribbon (DeBondt *et al.*, 1993). Between the two lobes is a 40 amino acid portion which constitutes the deep catalytic cleft with ATP phosphate, the substrate binding sites and the activation loop. The activation loop contains the phosphorylation site Thr160 (for human CDK2 and CDK5, for human CDK6 Thr177) and is specially called T-loop in CDKs (Fig. 2.4).

Sequential activation of members of the CDK family promotes the correct timing and ordering of events required for cell growth and cell division. In addition to driving progress through the cell cycle, CDKs are also the downstream targets of checkpoint pathways. These checkpoints act to ensure that critical cell cycle events have been successfully completed before the cell progresses into the next cell cycle stage. They are composed of a surveillance system that detects when a particular cell cycle event has not been correctly executed. Monomeric CDKs are inactive and require both association with the positive regulatory subunit, a cyclin, and phosphorylation on the conserved threonine residue that lies within the activation T loop for full activity. Only CDKs 1, 2, 4 and 6, when bound to their cognate cyclins, appear to have major roles in controlling cell cycle progression in humans (Johnson LN, 2002).

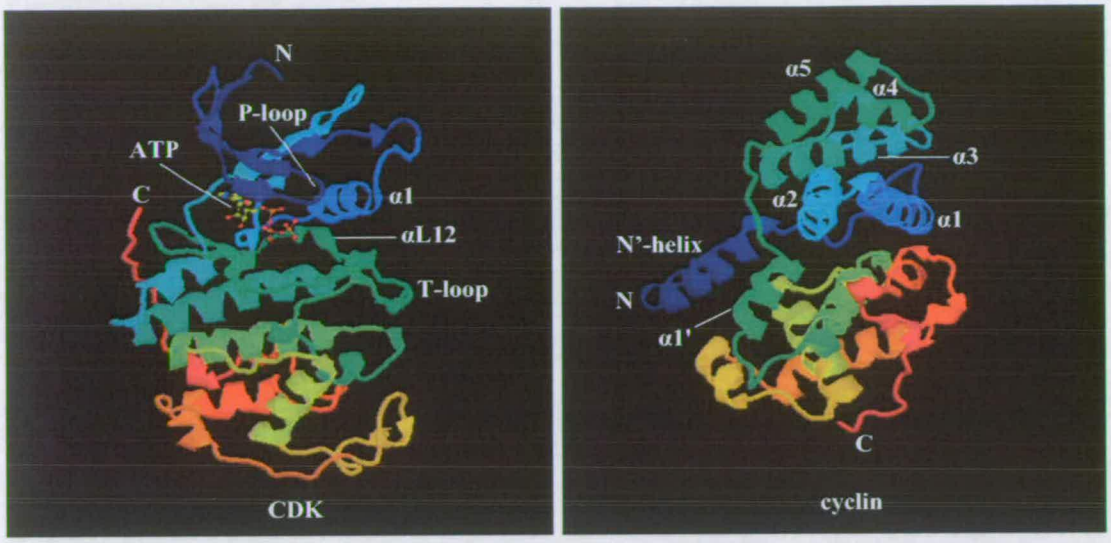


Fig. 2.4 Cartoon representations of catalytic domain of human CDK 2 and cyclin A. Both CDK2 and cyclin A are coloured gradually from blue (N-terminus) to red at C-terminus. Several important motifs in CDK are labelled. The diagram was created with RasMol (Sayle RA & Milner-White EJ, 1995).

Twelve paralogous CDKs and eleven paralogous cyclins have been identified in humans (Meyerson M, *et. al.*, 1999; Chen HH, *et al*, 2006; Grana X *et al*, 1994; Kikuno R, et al, 1999; de Graaf K *et al*, 2004). CDKs in other animals can be easily recognised by their sequence similarity to characterised CDKs and by the presence of a variation of the conserved "PSTAIRE" motif. Even before their associated cyclins are discovered (if any are associated), these "CDK-related kinases" can be classified following the sequence of their PSTAIRE motif: PSTAIRE for CDK1-3, PISTVRE for CDK4, PSSALRE for CDK5, PLSTIRE for CDK6, NRTALRE for CDK7, MSACRE for CDK8, PITALRE for CDK9 (Meyerson M, *et. al.*, 1999), PISSLRE for CDK10 (Grana X *et al*, 1994), and PITSLRE for CDK11 (Kikuno R, et al, 1999).

The archetypal cyclins A and B of animals share a conserved 250-amino acid domain called the cyclin core (Nugent JHA, *et al.*, 1991; O'Farrell P. and Leopold P, 1991). The cyclin core of cyclin A is sufficient for binding and activation of CDK1 and

CDK2 (Kobayashi H. *et al.* 1992; Lees EM and Harlow E., 1993). It displays a rigid tertiary structure organized in two structural repeats of five helices each (Fig. 2.4). The first repeat, running from Tyr199 to Leu306, encompasses the CDK-binding site (Jeffrey PD. *et al*, 1995). The first helix in this repeat, helix α1, is the most highly conserved in the cyclin family. Helices α2 and α3 are largely buried and form the core of the fold. The first structural repeat is the defining feature of all cyclins, and is called the cyclin box. It is conserved to varying degrees in all cyclin classes (Brown NR., *et al.*1995). Although the cyclin superfamily is diverse, the percentage sequence identity between cyclins that regulate G1 and G2-M can be fairly low even within the cyclin box.

2.9.2 CDK Regulation

The crystal structure of free human CDK2 (DeBondt *et al.*, 1993) and the human CDK2-cyclin A complex (Jeffrey PD. *et al*, 1995) are helpful to understanding the structural details of the activation of CDK2. In the inactive CDK2 apoenzyme, the T-loop blocks access to the catalytic site between the two lobes. The CDK2-cyclinA interface involves the α1 helix (PSTAIRE helix), the T-loop and portions of the N-terminal β-sheet and of the C-terminal lobe of CDK2, and helices α3, α4 and α5 of the cyclin box, and the N-terminal helix, of cyclin A (Figure 2.6A).

On CDK2, the most extensive binding to cyclin A occurs within the α1 helix. Binding induces translation of this helix into the catalytic cleft towards the ATP and a rotation of roughly 90^0 along its helical axis into the catalytic cleft. This leads to tighter packing in cyclin-bound CDK2 compared with free CDK2 and proper aligning of active site residues such as Glu51. Extensive binding also occurs between

cyclin and the T-loop. This induces the "melting" of the αL12 helix in CDK2 at the beginning of the T-loop, and relieves the block αL12 helix exerts on the catalytic cleft, allowing the α1 helix to move deeper into the catalytic cleft. At the same time, the β-strand that replaces the αL12 helix in the CDK2-cyclin A complex structure directs the T-loop to move further away and relieve the steric block from the entrance of catalytic cleft (Fig. 2.6). But only phosphorylation of Thr160 in CDK2 induces the fully active T-loop conformation (Jeffrey P.D. *et al.*, 1995). This phosphorylation is catalyzed by CDK-activating kinase (CAK) in association with a third protein, MAT1 (for 'menage a trois-1'). The CAK might itself consist of CDK7 complexes with cyclin H. The CDK subunit must also be dephosphorylated on two residues, Thr14 and Tyr15 in CDK2, located in the P-loop at the border of the ATP-binding pocket. P-loop typically contains a conserved glycine rich motif which makes it very flexible in the absence of ATP (Walker JE *et al.*, 1982; Huse M. and Kuriyan J., 2002).

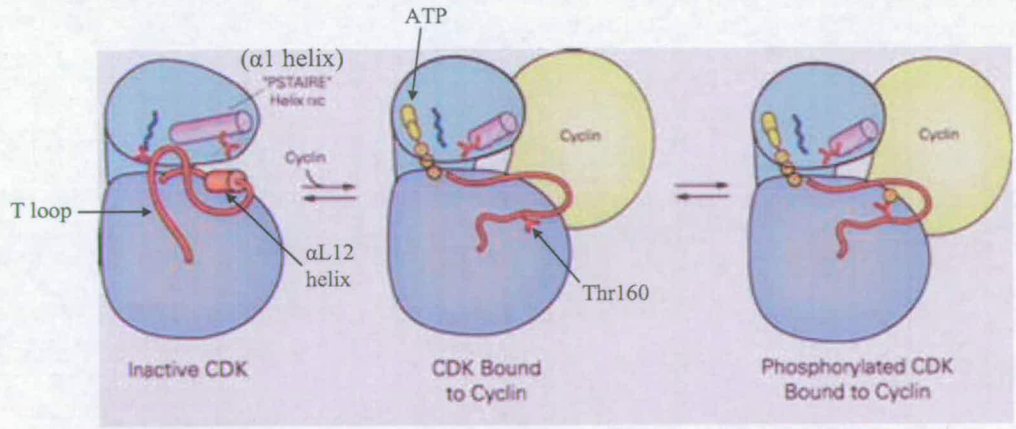


Fig 2.5 The regulation of CDK. This cartoon representation of the activation procedure of CDK was modified from reference Huse M. and Kuriyan J., 2002.

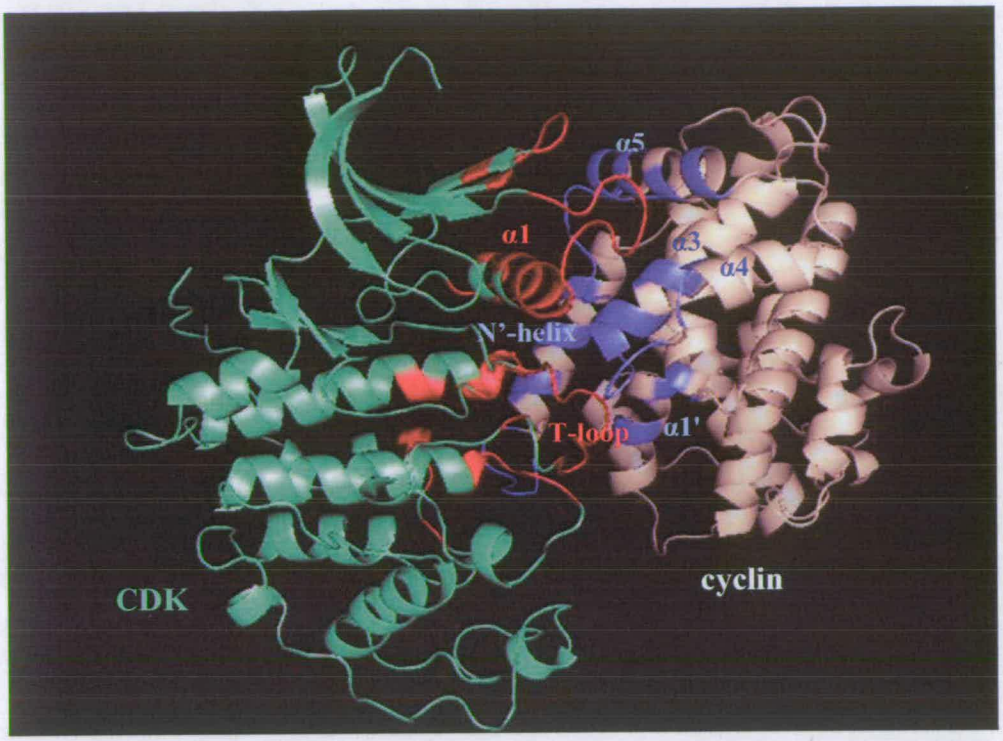
In contrast to the activating phosphorylation by CAK, CDK1 and CDK2 are also regulated by inhibitory phosphorylation of a tyrosine residue near the N-terminus

(Tyr 15 on human CDK1/2) catalyzed by the Wee 1 protein kinase. This inhibitory phosphorylation can be reversed by members of the CDC25 family of protein phosphatases (Nurse P, 1997).

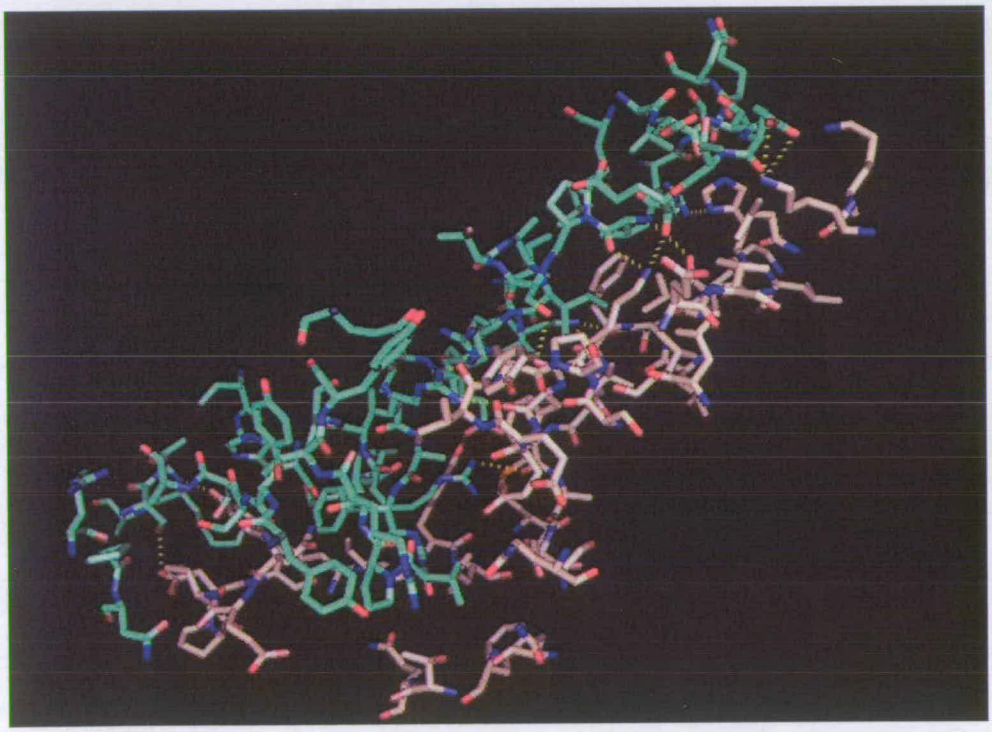
Finally CDK activities can also be controlled by the binding of CDK inhibitors to the CDK-cyclin complexes. In mammalian cells, there are two main families of CDK inhibitors. Members of the Cip/Kip family members regulate all stages of progression through G_1 and S phase while Ikn4 family members only regulate progression through the check point in G_1. These CDK/cyclin complexes are then additionally controlled by mechanisms that include inhibitory phosphorylation, protein association, sub cellular localisation and targeted destruction of regulatory proteins (Johnson LN, 2002).

2.9.3 Main Interactions between Human CDK2 and cyclin A

Contacts between human CDK2-cyclinA 2 mainly involve hydrogen bond networks and hydrophobic interactions. The majority of these contacts are seventeen inter-molecule hydrogen bonds (including hydrogen bond formed between main-chain atoms) (table 2.2). Hydrophobic interactions are less extensive than hydrogen bond networks. No obvious hydrophobic patches can be observed on the interface of CDK2-cyclin A2. We found twenty nine hydrophobic residue pairs out of the 172 inter-chain contacting residue pairs in the structure of CDK2 and cyclin A2. Similar contact properties are observed in human CDK6-viral cyclin (Jeffery PD *et al*, 2000) and CDK5-P25 (Tarricone C, *et al.* 2001).



A



B

Figure 2.6 Human CDK2-cyclin A2 Interactions. A: Secondary Structures of CDK2 and cyclin A2 involved in interactions. CDK2 is highlighted in lime, cyclin in wheat, interface

residues in red (on CDK2) and slate (on cyclin A2). Figure B: Inter-molecule Hydrogen Bonds between human CDK2 and cyclin A2. Inter-molecule contacting residues of CDK2-cyclin A2 are represented in stick style, Oxygen atoms in red, Nitrogen in blue, Sulphate atom in orange, Carbon in green (CDK2) and wheat (cyclin A2), hydrogen bonds are represented as golden dashes. The diagram was created with PYMOL (DeLano Scientific LLC).

	Inter-molecule Contacting Residues
CDK2	Leu37, Asp38, Thr39, Glu40, Thr41, Glu42, Gly43, Val44, Pro45, Ser46, Ala48, Ile49, Arg50, Ile52, Ser53, Leu54, Lys56, Glu57, Leu58, Val69, His71, Thr72, Glu73, Asn74, Leu76, Leu115, Ala116, His119, Ser120, His121, Arg122, Val123, Leu124, Arg150, Ala151, Phe152, Gly153, Val154, Pro155, Val156, Arg157, Thr158, Tyr159, Glu162, Tyr179, Tyr180, Ser181, Thr182, Ala183, Pro271, Asn272, Lys273, Arg274, Ile275, Ser276, Ala277, Lys278, Ala279
Cyclin A2	Asn173, Glu174, Val175, Pro176, Asp177, Tyr178, Asp181, Ile182, Tyr185, Leu186, Met189, Gln228, Glu230, Leu262, Leu263, Ser265, Lys266, Phe267, Glu268, Glu269, Ile270, Tyr271, Pro272, Glu274, Val275, Lys288, Lys289, Leu292, Glu295, His296, Ile297, Leu299, Lys300, Leu302, Thr303, Phe304, Asp305, Leu306, Ala307, Ala308, Asn312, Gln313, Phe314, Thr316, Gln317
Reside Pairs (R_{CDK2}-$R_{cyclinA2}$) which form HBs	Glu40-Lys288, Glu42-Lys266, Val44-Lys266, Val44-Glu295, Ser46-Lys266, Lys56-Thr303, Lys56-Tyr185, Glu57-Tyr185, His71-His296, Arg122-Ala307, Arg122-Ala307, Arg150-Glu269, Arg150-ILE270, Val156-Asn173, Ser276-Asp177, Lys278-Tyr178, Lys278-Asp181.

Table 2.2 Inter-molecule contacts of human CDK2-cyclin A2 (pdb entry name 1FIN). If the distance of two residues coming from different chains of a complex structure is less than 6 Å, these two pairs are defined to be in contact. The calculation was done by perl script (Appendix C). Hydrogen bonds (HB) are calculated by Protein-Protein Interaction Server (*http://www.biochem.ucl.ac.uk/bsm/PP/server*).

2.9.4 Plant CDKs and cyclins

2.9.4.1 Plant CDK Nomenclature

A previous structurally grounded classification that is also consistent with the functional characteristics of plant CDKs divides them into five classes (Joubès J, et al, 2000). The CDKA family is the group bearing the PSTAIRE motif, or a slightly altered PSTALRE motif in *Dictyostelium discoidum* CDC2 homologues. This CDK group regulates both the G1-to-S and the G2-to-M transitions of the cell cycle. Among a total of 38 conserved residues (with respect to all animal and yeast CDKs) which are exposed to solvent in free human CDK2, the CDKA in *Arabidopsis* has only three amino acid changes compared with human free CDK2 (Joubès J, et al, 2000). This suggests a high degree of functional similarity within the PSTAIRE group of CDKs. However, the cyclin binding partner of plant CDK A remains unclear even through some reports using different methods indicate that plant CDK A can bind plant cyclin D (Deveylder *et al.*, 1997; Nakagami *et al.*, 1999).

The plant CDKB family is a plant-specific, evolutionary conserved gene class with PPTALRE or PPTTLRE or PSTTLRE at the motif site. The two consecutive prolines constitute a plant-specific hallmark among CDK-related kinases (Burssens *et al.*, 1998; Mironov *et al.*, 1999). The CDKB group seems to control the G2-to-M checkpoint only (Porceddu *et al.*, 2001). There are 16 amino acid changes between the CDKB in Arabidopsis CDKB and human free CDK2, amongst the 38 conserved residues (with respect to all mammal and CDKBs) exposed to solvent in free human CDK2 (Joubès J, et al, 2000). There are no data available about the cyclin partner of plant CDKB, although there is some supposition that plant CDKB may bind to plant cyclin B which is expressed at the same time-point as CDKB in the cell cycle.

The plant CDKC family is characterised by the presence of a PITAIRE motif, which is also present in the human CDK-related CHED kinase family. This family also includes the human CDK9 kinase which has a PITALRE motif (Lapidot-Lifson *et al.*, 1992; Defalco and Giordano, 1998). No data is available about the function, cyclin partner and substrate of this kind of CDK in plant up to now.

The plant CDKD family has a N(I/V/F)TALRE motif and corresponds functionally to human CDK7. One rice CDKD kinase, R2, displays the same functional activity as human CDK7 as it is able to phosphorylate not only the rice CDKA but also human CDK2 and RNA polymerase II from *Arabidopsis* in vitro (Yamaguchi M., *et al.*, 1998).

The plant CDKE family harbours a unique SPTAIRE motif, and seems to be unrelated to any other groups of plant sequences. The most similar, although still only distantly related, protein in the databased, is human CDK8 which has the SMSACRE motif and is involved in the regulation of RNA polymerase II together with partner cyclin C (Magyar *et al.*, 1997).

In *Arabidopsis*, two functional classes of CDK-activating kinases (CAK) have been identified. One class, includes CDKD;1, CDKD;2 and CDKD;3, which display 75%, 68%, and 79% sequence similarity to rice CAK, R2 respectively, at the protein level. Another class, cak1At, differs substantially from R2 and was renamed CDKF;1 with the greatly altered motif A---FRE ('-' means gap) (Vandepoele K. *et al.* 2002). CDKF,1 is plant-specific and shows cyclin H-dependent CDK-kinase activity. Murray JAH group (Menges M et al, 2005) identified two new CDK-like sequences in *Arabidopsis* which harbour a PLTSLRE motif and named them as CDKG;1 and

CDKG;2. These two CDK homologues seem to control the G1-to-S checkpoint of *Arabidopsis* cell cycle.

2.9.3.2 Plant Cyclin Nomenclature

The majority of plant cyclins described to date display similarity with animal cyclin classes A, B, D, H. Therefore, plant cyclins were named with the mnemonics CycA, CycB, CycD, CycH, respectively, to indicate their sequence relationship to the equivalent cyclins in other eukaryotes (Renaudin JP, *et al.*, 1996; Yamaguchi M, *et al*, 2000.). These three groups were divided up further into clusters: CycA1, CycA2, CycA3, CycB1, CycB2, and CycB3, are mitotic cyclins; CycD1, CycD2, CycD3, CycD4, CycD5, CycD6 and CycD7, are G1-specific cyclins. The common signature of all A-type cyclins is a LVEVxEEY (x = any amino acid) motif locating on the α1 helix. Plant B-type cyclins have a (H/Q)x(K/R?Q)(F/L) motif on the α1 helix. Another typical signature of yeast and animal B type cyclins, FLRRxSK on the α1' helix, is also found in all CycA and CycB cyclins in plants but in an altered form, lacking the serine. Plant CycD retains the signature of animal cyclin Ds, the retinoblastoma protein (Rb)-binding motif LxCxE locating on the N-terminus of the proteins, though the overall pair-wise sequence identity between *Arabidopsis* CycD and animal cyclinD is only 9-14% even in the cyclin box region, the most conserved region of the cyclins.

Rice CDKD kinase, R2, does not have kinase activities as a monomer, and needs to bind to another regulator subunit to be activated (Yamaguchi M., *et al.*, 1998). In poplar (*Populus tremula X tremuloides*) and rice (*Oryza sativa*), cDNAs encoding cyclinH homologues have been detected (Yamaguchi M. *et al*, 2000). Both of these,

named Pt;cycH;1 and Os;cycH;1, are able to interact with rice R2 and regulate the kinase activities of this rice CDKD kinase. Os;cycH;1 accumulated during S phase in partially synchronized suspension cells.

2.9.5 *Arabidopsis thaliana* CDK and cyclin

2.9.5.1 *Arabidopsis thaliana*

Arabidopsis thaliana is a small flowering plant belonging to the mustard family which includes cultivated species such as cabbage and radish. *Arabidopsis* is not of major agronomic significance. However it offers important advantages for basic research in genetics and molecular biology:

- a small genome with extensive genetic and physical maps of all five chromosomes;
- a short life cycle (from germination to the mature seed in only about six weeks) and easy cultivation in restricted space;
- efficient transformation methods using *Agrobacterium tumefaciens*;
- plenty mutant lines and genomic resources.

Such advantages have made *Arabidopsis thaliana* a model organism for studies of the cellular and molecular biology of flowering plants *(http://www.arabidopsis.org/info/aboutarabidopsis.jsp)*.

The *Arabidopsis thaliana* genome, ~125 Mbp in length, was sequenced and annotated before the end of 2000 (*Arabidopsis* Genome Initiative, 2000). Knowledge of the complete sequence of *Arabidopsis* is directly relevant to human biological studies, because many fundamental life processes at the molecular and cellular levels are common to all higher organisms. Some of these processes are easier to study in *A.*

thaliana than in human or animal models. *A. thaliana* contains numerous genes equivalent to those that prompt disease in humans -- ranging from cancer and premature aging, to ailments such as Wilson's disease, in which the human body's inability to excrete copper can be fatal.

2.9.4.2 CDK and cyclin like sequences discovered in *A. thaliana*

Joubes and Boudolf reported one A-type and four B-type CDKs in *A. thaliana* (Joubes J., 2000; Boudolf V. *et al*, 2001). Using homology-based annotation methods on the whole *Arabidopsis* genome, Vandepoele K. identified one A type, four B type, two C type, three D type, one E type, and one F type CDK-like sequence (Vandepoele K, *et al*, 2002). Previous work in Doerner group identified 35 CDK like sequences in *A. thaliana* through transitive BLAST searches (A.Molesworth & P. Doerner, unpublished results). The twenty previously unclassified new CDK homologues can be divided into two clusters: five with a A(L/K)RTLRE motif; and a second cluster which is a special class. Interesting, all the 15 closely-related sequences in the second cluster share a "MGC", sometimes "MGCIC" N-terminal motif. Moreover, all of them, (except At4g22940 for which it is IKCIARE), have a V(K/R)FMARE motif in the location corresponding to the human CDK2 PSTAIRE motif. On the phylogenetic trees of all 35 CDK-like sequences, these 15 sequences cluster together and form a distinct branch from the others (Fig.6.1, 6.2 in Chapter 6). All these features indicate that this might be a novel plant CDK-like protein group.

To date, 47 cyclin homologues have been reported in *A. thaliana,* ten A-type cyclins, eleven B-type cyclins, two C-type cyclins, ten D-type cyclins, one H-type cyclin, five T-type cyclin, seven P-type cyclin, and one L-type cyclin (Vandepoele K, *et al*,

2002; Torres A *et al*, 2004; Barroco RM et al, 2003; Wang G *et al*, 2004; Menges M *et al*, 2005). Here also previous work in the Doerner group has revealed another 3 cyclin homologues in *A. thaliana* (A. Molesworth & P. Doerner, unpublished results).

3.

Project Principal Aims

The transient molecular interactions between the cyclins and their CDK partners are amongst the most relevant regulatory events in eukaryotes. The genome sequencing projects are revealing a large number and variety of CDK and cyclin sequences. To date a lot of bioinformatics research is undertaken on protein-protein interactions by various research groups. However, there are still no systematic computational methods available to predict specific protein-protein interactions between sets of paralogous proteins, e.g. which CDK-cyclin pairs can form transient complexes, and which do not.

The general aims of this project were to investigate, the biochemical and biophysical characteristics and principles that determine the specificity of known protein-protein interactions in order to discover new ones, and to develop a novel computational protocol by combining multiple components from several existing methods that can be applied generally in this field. The content of this project will be:

- ✓ To develop a strategy that combines comparative modeling, all-by-all docking, and re-ranking selection criteria to predict specific transient protein-protein interactions. This strategy will focus on predicting likely interactions between CDKs and cyclins.
- ✓ To test this prediction approach where possible
- ✓ To estimate the prediction accuracy
- ✓ To apply and validate this prediction approach

Previous studies and sequence sub-classification of the cyclin and CDK multi-gene families in the genome of the model plant *Arabidopsis thaliana* have revealed a

minimum of 50 cyclin-like, and 35 CDK-like putative gene products. Identification of potential CDK-cyclin pairing between these will be of particular interest in this project.

4.

Standard Methods

4.1 Strategy Overview: Large Scale Docking + Selection Criteria

Molecular docking is the only tool currently available to predict specific protein interacting partners. Here we used the comparative protein structure modelling method to build model structures of *A. thaliana*, *Trypanosomatid brucei*, *Leishmania major*, and human CDK and cyclin homologues. Then these CDK and cyclin model structures were subject to a large scale docking with ZDOCK in which all CDK-cyclin combinations were considered. However, automatic molecular docking results usually contain too many false positives to be used directly. We therefore applied a set of additional criteria to select the best CDK-cyclin docked complexes from the ZDOCK result. These selection criteria include: ZDOCK score and associated Z score; relative orientation of CDK/cyclin subunit in the docked complexes; and interface surface property criteria.

4.2 CDK/cyclin Structures in PDB

'Cyclin' was used as a keyword to run a full text search of the Protein Data Bank (PDB) database (Westbrook *et al.*, 2002). A non-redundant set of structures of CDK/cyclin were selected. When there are several structures available for one protein, the structure with the best resolution was chosen (table 5.1). These structure files were modified either to remove extra copies of chains or ligands or to add missing residues and atoms using the program SYBYL ((Tripos Inc.).

PDB entry	Proteins	Source	Resolution
1FIN	CDK2 + cyclin A2	*Homo sapiens*	2.2 Å
1F5Q	CDK2 + M-cyclin	*Homo sapiens,* Murine γ-herpesvirus	2.5 Å
1G3N	CDK6 + K-cyclin + Ink4	*Homo sapiens,* Sarcoma-associated herpesvirus	2.9 Å
1JOW	CDK6 + V-cyclin	*Homo sapiens, Herpesvirus saimiri*	3.1 Å
1H4L	CDK5 + P25	*Homo sapiens*	2.65 Å
1JKW	Cyclin H	*Homo sapiens*	2.6 Å
1B38	CDK2	*Homo sapiens*	2 Å
1BLX	CDK6	*Homo sapiens*	1.9 Å

Table 4.1 CDK and cyclin structures available in PDB.

Three human CDK (CDK2, CDK5, CDK6) and 3 human cyclin subfamilies (cyclin A1-3, P25, cyclin H) have appropriate structures in PDB (Table 4.1). Complexes structures that are available are: CDK2-cyclin A2, CDK5-P25, CDK2-M cyclin (D-type viral cyclin), CDK6-V cyclin (D-type viral cyclin) and CDK6-K cyclin-Ink4 (K-cyclin is a D-type viral cyclin).

CDK2 and CDK5 in 1FIN and 1H4L can be treated as in similar conformations—the transient conformation of being bound with cyclin but not phosphorylated (CDK5 might have a slightly different activation mechanism as the binding of P25 alone can stabilize

its active conformation (Jeffrey P.D. *et al.*, 1995, Tarricone C. *et al*, 2001)). Therefore they can be used as template structures for modelling the interaction of other CDK homologue with cyclins. Molecular docking experiments using these models can also be regarded as "bound docking".

Viral cyclins are unbeneficial for the host and are therefore under different evolutionary pressure. In the CDK2-M-cyclin structure, the viral cyclin binds CDK2 with an orientation slightly different from cyclin A though it activate CDK2 by triggering conformational changes in a similar way to cyclin A (Card G.L., *et al* 2000). The complex structure of CDK6-Vcyclin is similar to phosphorylated CDK2-cyclinA structure and displays resistance to inhibition by INK-type CDK inhibitors. In the CDK6-Kcyclin (D-type viral cyclin) complex structure the inhibitor Ink4 distorts both the ATP binding site and the cyclin-binding site, weakening the cyclin's affinity for the CDK. Hence both these two CDK6 conformations are not suitable for direct use as template structures.

4.3 Evolutionary Trace

The Evolutionary Trace (Lichtarge *et al,* 1996) is a systematic functional interface prediction technique, which is based on two observations and two extended hypotheses: Observation one is that protein structures are conserved amongst homologous proteins. Another observation is that active site residues undergo fewer mutations during evolution than less functionally important residues due to evolutionary pressure. Based on these two observations, the authors of the program proposed two extended hypotheses: One is that evolutionarily-conserved residues are functionally more

important than evolutionarily-variant residues. The second hypothesis is that a sequence identity dendrogram reflects appropriate functional classification. The program extracts functionally important residues from sequence conservation patterns within a family of homologous proteins, and then maps them onto the protein 3D surface to identify clusters that may characterise functional interfaces.

The input for ET consists of a protein family with divergently related sequences in a multiple sequence alignment and a derived sequence identity dendrogram. The protein family is partitioned into sets of an increasing number of subgroups that are delineated by branch points in the dendrogram, beginning with one group containing all of the sequences in the family and ending with each protein being its own subgroup. The "evolutionary rank" of a residue is the minimum number of branches into which the dendrogram must be divided for it to become a trace residue. As the rank increases, class specificity is linked with increasingly minor evolutionary divergences. The final step in ET is to map the top-ranked residues onto the structure and then to assess whether they are spatially clustered.

In our analysis, the standard Blast tool from the NPSa web server (http://npsa-pbil.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_server.html) was used to gather sequence fragments that match the protein domains of interest. For the CDK domain, Blast searches of the Swiss-Prot (Bairoch A, *et al*, 2004) database were carried out using human CDK2, CDK5 and CDK6, respectively, as query sequences. Human cyclin A, K-cyclin and P25, respectively, were used for the analogous searches for cyclin domain homologues. The list was truncated where the proteins retrieved displayed E-values higher than 10^{-15} and/or when their function became clearly unrelated to CDK/cyclin.

Sequence alignment, dendrogram construction and evolutionary trace analysis were then carried out with the Binding-Site Analysis module of INSIGHT II (Accelry Ltd.). All sequences were reordered according to the dendrograms and then submitted to DARWIN web server (Gonnet GH, *et al*, 2000) to build phylogenetic trees. The dendrograms were compared with the phylogenetic trees in order to refine the positions of each sequence.

4.4 Comparative Modelling

None of the *Arabidopsis thaliana* 35 CDK and 50 cyclin homologues' 3D structures have been solved experimentally. To run the large scale docking approach to predict their interactions, we adopted a comparative protein structure strategy to build their 3D structure models.

The general comparative modelling procedure consists of four steps (MartÃ-Renom, M.A., Yerkovich, B. and Sali A., 2002) (1) Identification of known structures that are related to the target sequence to serve as templates. This step is facilitated by running BLAST search against the structure database PDB. The template candidate list was identified by E-value lower than 10^{-20} for CDKs and 10^{-5} for cyclins. Then a thorough check of protein annotation was carried out to make sure these proteins are CDK or cyclin.

Once a list of all related protein structures is obtained, the most appropriate template was selected. Usually higher global sequence identity between the target and template sequences yields a better model. Here we selected the template structures based on the PAM distance between target sequence and template sequence on the phylogenetic trees

built by DARWIN server. Several other factors were also taken into account (a) Subfamily factor: if the list of sequences, (including the target sequence), can be divided into subfamilies, the sequence from the sub-family that was closest to the target sequence was selected. (b) 'Environment'. The template environment should be compared to the required target structure environment. The term environment includes factors that determine protein structure, beside its sequence (e.g., solvent, pH, ligands, and quaternary interactions). In our case, because of the involvement of conformation changes of CDKs during activation or inhibition, only the structures of CDKs which are bound to appropriate cyclin partner but which are neither phosphorylated nor inhibited were used. (c) The quality of the experimental template structure. When several experimental structure models are available for one template sequence, the one with the highest resolution is selected.

(2) The second step, which is also the most important step in comparative modelling, is to align the target sequence with the selected template(s). There are a great variety of protein sequence alignment methods, many of which are based on dynamic programming techniques. Here we used the best tools currently available to generate our alignment. The sequence identities between CDK homologues are generally quite high and T-Coffee (Notredame C, *et al*, 2000) was used to build their alignment.

Hidden Markov Model based alignment methods are generally viewed to be more powerful than conventional methods when very distant homologues are to be detected and aligned (Orengo CA, *et al*, 2003). SUPERFAMILY (Gough J, *et al*, 2001) and PFAM (Bateman A, *et al*, 2004) are two large databases of protein domain families and contain curated multiple sequence alignments for each family. Pfam is a large collection

of multiple sequence alignments and hidden Markov models covering many common protein domains based on the Swiss-Prot/TrEMBL (Watanabe K and Harayama S, 2001) protein sequence databases. The profile HMM is built from the seed alignment using the HMMer package which is then used to search the pfamseq sequence database. All the matches found above the curated thresholds are aligned using the profile HMM to make the full alignment. The purpose of SUPERFAMILY is to provide structural (and hence implied functional) assignments to protein sequences at the Structural Classification of Proteins (SCOP) (Murzin AG, *et al*, 1995) superfamily level. SCOP provides a classification of all proteins in the Protein Data Bank (PDB). Whole genome assignment results are also provided. The SUPERFAMILY server is based upon the SAM (Hughey R & Krogh A, 1996; Karplus K et al, 1998) HMM software, and release 1.67 of the SCOP structural classification of proteins.

We used SUPERFAMILY to generate the alignment of human and *Arabidopsis* cyclin homologues. PFAM alignments of human cyclin family were extracted and manually edited to be used in our further work.

(3) Model building: Once an initial target-template alignment has been built, a variety of tools can be used to produce a detailed 3-D model for the target protein. In our experiment, we use MODELLER 6 version 2 by Sali Blundell *et al.* (Sali A., Fiser A., *et al.* 2001).

(4)The final step of comparative modelling is model evaluation. The quality of a model determines whether the information extracted from it will be reliable. For example, MODELLER automatically gives a score relating to the perceived quality of the model structure. This score is named MODELLER's "objective function" and is reported in the

second line of the model PDB file. In our experiment, ten models were built with MODELLER for each target sequence. The model with the lowest objective function was selected for further use.

A set of model quality evaluation methods, ERRAT (Colovos C and Yeates TO, 1993), PROVE (Pontius J *et al.*, 1996), PROCHECK (Laskoswki RA, *et al.*, 1993), and Verify_3D (Luthy R., *et al.*, 1992), were also used to evaluate the different structural quality aspects of these models.

Procheck analyzes the stereochemical quality of a structure and produces a number of PostScript plots analyzing its overall and residue-by-residue geometry (Laskoswki RA, *et al.*, 1993). The proportion of residues in core and allowed region in the Ramachandran plots calculated by Pro-check evaluate the stereochemical quality of models. The proportion of residues in most favoured regions plus residues in additional allowed regions is an important index of model quality. If this percentage is above 90%, the model quality is good and there are not many stereo-chemical clashes. The program Verify_3D assesses protein structures with three dimensional profiles (Luthy R., *et al.*, 1992). This program calculated the statistical preferences, termed as 3D-1D scores, of each of the 20 amino acids for the environment of each residue position in the 3-D model. The environment parameters of each residue position are the area of the residue that is buried, the fraction of side-chain area that is covered by polar atoms (O and N); and the local secondary structure. Using the averaged data points produced for each amino acid in the sequence, the numbers of times the value is less than 0.2 is converted into the percentage that has positions with values less than 0.2. If this percentage is larger than 80%, the model quality is satisfactory. If the percentage is between 80% and

55%, the model might need adjustment. If the percentage is lower than 55%, adjustment is strongly recommended. ERRAT (Colovos C and Yeates TO, 1993) plotted error values as a function of a sliding 9-residue window. The error function is based on the statistics of non-bonded atom-atom interactions in the reported structure by comparing to a database of reliable high-resolution structures. ERRAT gives the percentage of the sequence that is above the 95% confidence limits for each chain. The PROVE (Pontius J *et al.*, 1996) program calculates a number of derivative scores from standard atomic volumes. The standard atomic volume was computed in 64 crystal protein structures (using Voronoi procedure) for use as a quality measure for protein structures. One example derivative score is described as the Volume-z score RMSD (root mean square deviation) to represent the R-factor of a crystal structure. In the PROVE evaluation, the percentage of the number of buried "outliers" is calculated. The "outliers" were defined in Prove to be the structures with volume-Z score exceeding their limits; they were treated as outliers. The limits were derived from Z score r.m.s distribution. If the percentage is less than 1%, the model quality is satisfactory.

4.5 Molecule Surface Potential Representation

Docking essentially simulates the interaction of the protein surfaces. Surface properties, for example electrostatic potential, play key roles in transient protein-protein recognition. Here we used GRASP (Nicholls A., *et al.*, 1991) to compute and display polyhedral molecular surface models of protein structures representing the electrostatic potential.

4.6 Protein Structural Comparison

The structural comparison program we used, QUANTA (Accelrys Inc.), employs rigid body methods to superpose equivalent Cα atoms between protein structures. First, the protein sequences were automatically aligned. Then the program tried to overlay the Cα atoms within the matched residues of the active structures using a least square algorithm. Program QUANTA runs several cycles of superposition to superpose multiple molecules. In the initialization cycle, the first selected molecule is set as target molecule and all the other molecules are superposed onto it one by one. In the following cycles, the program calculates an average structure from all the molecules in each cycle. Then all these molecules are superposed onto this average structure, and the RMSD difference in atomic coordinates between each molecule and the average structure is also calculated. This cycle is repeated until the RMSD difference in the average structure coordinates between cycles is less than 0.1Å. If convergence is not achieved, iteration is terminated after 10 cycles.

After the structures are superposed the RMSD between all the Cα atoms on different molecules are calculated and used as criteria to measure the structural similarity of these molecules.

4.7 Large Scale Molecule Docking

In our all-by-all docking approach, we mainly use ZDOCK (Chen R, et al., 2003; Chen R & Weng ZP, 2002). ZDOCK is an initial-stage protein docking algorithm based on Fast Fourier Transform (FFT). With the receptor fixed at the original position, the

algorithm searches the entire rotational and translational space of the ligand with respect to the receptor at a 6° rotational sampling interval around each of its three Cartesian axes. For each rotation angle, the algorithm rapidly scans the translational space using FFT. The ZDOCK scoring function combines the following components: pair-wise shape complementarity, desolvation and electrostatics. ZDOCK was around 90% accurate on a benchmark of 49 non-redundant test cases, including antibody-antigen and enzyme-inhibitor complexes (Chen R, et al., 2003).

ZDOCK has the potential to exclude non-interface residues from the comparison and only tries different translational and orientational position around the binding site. We wrote Perl scripts to extract the non-interface residue list based on target-template alignment and template complex structures. Residues in one chain of a complex were defined to be in contact when their side-chain non-hydrogen atoms are within 5Å of the side-chain non-hydrogen atoms of any residue in another chain in this complex. To give ZDOCK a certain freedom in searching, we also leave the flanking four residues of interface residues on the alignment unmasked. Large scale docking with ZDOCK was also automated using a perl script.

4.8 Selection Criterion Calibration

Automated protein-protein docking results typically contain too many false positive complexes to be directly useful in practice. We therefore applied additional criteria to select the best complexes from the ZDOCK result lists.

4.8.1 ZDOCK score and z score

The ZDOCK score may reflect how tightly the two proteins are docked together. The highest ZDOCK score of each pair and their corresponding z score are used to divide all the output pairs into those with a high docking affinity from complexes and those with a weak docking affinity.

4.8.2. Subunit Orientation

Each pair's docked complex structure was superposed with the two crystal human CDK-cyclin complex structures (pdb entry name 1FIN and 1H4L). Cα atom RMSD between docked complex structures and the two human complex structures were calculated using the QUANTA program. The Cα atom RMSD was calculated as a criterion of subunit orientation.

4.8.3. Interface Property Criterion

4.8.3.1 MolSurfer Coefficients

Electrostatic and hydrophobic interactions are considered to be an important factor in stabilizing protein-protein complexes. MolSurfer (Gabadoulline *et al.*, 2003) is a Java program that can project three-dimensional interfaces of protein-protein complexes to two-dimensional maps and calculate correlation coefficients of surface properties on interfaces. MolSurfer interface electrostatic correlation coefficient (ECC) and residue-residue hydrophobic correlation coefficient (HCC) were used in our work as interface criteria to select potential CDK-cyclin complexes.

We downloaded the stand alone version of MolSurfer and modified it to do following things: 1) calculate the number of pixels it divided each interface map into. This pixel number provides a rough estimate of the interface area; 2) write the pixel number and all surface property correlation coefficients to a text file so that we can run all-by-all MolSurfer automatically.

To make MolSurfer work, charges need to be assigned to each atom in pdb files. The standalone program PDB2PQR (Dolinsky TJ. *et al*, 2004) did this work using the AMBER force field. Output of PDB2PQR works as input files for APBS (Baker NA, *et al*, 2001), which is a Poisson-Boltzmann Equation (PBE) solver, to create .grd files. Finally, the pdb files and grd files for each chain in a complex, plus the pdb file of the complex, were input into the stand alone MolSurfer to calculate electrostatic and hydrophobic correlation coefficients. All these processes were automated by means of perl scripts.

4.8.3.2 Reference Sets

To calibrate the interface property criteria, we need to generate two reference sets: The positive set consisted of transient hetero-dimer complexes and the negative set of non-complexes. The original transient heterodimer complex PDB entry name and the contacting residue numbers on interfaces between chains in each PDB file come from Yanay Ofran (Ofran & Rost, 2003). The sequences of these proteins are extracted from PDB using the SRS7 system server (Zdobnov EM, *et al.*, 2002). The following criteria were used to select chain pairs for further work: (a) chain length $\geq$ 60 residues, (b) contact area size (measured as the solvent inaccessible surface area calculated by the

Protein-Protein Interaction Server (Jones S. & Thornton J.M., 1995)) ≥ 600 Å^2, (c) structure integrality (no missing residues/atoms), and (d) PAM distances between sequences (calculated by DARWIN server (Gonnet GH, *et al*, 2000)) larger than 130.

The negative set, "non-complexes", was generated by using ZDOCK to combine proteins that normally do not interact. Transient hetero-dimer complexes coming from positive control set and non-homologous obligate complexes coming from Ofran & Rost's dataset were split into two chains. All-by-all docking between these chains was run to create non-complexes. Selection criteria were (a) docked structures with ZDOCK score ≥ 60, and (b) interface inaccessible surface area ≥ 600 Å^2.

3.8.3.3 Discriminant Function Analysis

Both these two sets have two variates, ECC and HCC. Here we carried out discriminant function analysis with MATLAB (The MathWorks Inc) to generate new canonical functions, CEH1 and CEH2, which are linear combinations of the mean-centered original variables, ECC and HCC.

Cutoff values were decided computationally automatically by select the crossover point of the Gaussian distribution curves of the CEH1 values of the two control sets.

To evaluate the separation accuracy of this criterion, a cross-validation was carried out by randomly selecting 80% of data from each control set to derive the parameter values of discriminant function analysis and CEH1 cut-off value. Then these values were applied to calculate CEH1s for the other 20% complexes and also predict these

complexes. The same computation was repeated to calculate the separation percentage and the stability of the separation.

4.9 COMPUTATIONAL ENVIRONMENT

4.9.1.Procedures running on Silicon Graphics Origin200 server under IRIX 6.5, using two MIPS R12000 360 MHz processors:

A. Comparative Modelling: MODELLER 6 version 2.

B. Evolutionary Trace: Use the Binding Site module of INSIGHT II.

Surface Creation: SYBYL and GRASP.

MolSurfer: standalone MolSurfer with some source files modified and recompiled to be able to run at large scale. Large scale MolSurfer was automated by perl scripts.

4.9.2 Procedures running on DSZOE server: Alpha, 256Mb, RedHat 7.0.

Multiple Alignment: using T-Coffee and manual editing the alignment to try to move insertion/deletions out of secondary structure region.

4.9.3 Procedures running under on Linux-PC and CYCN: AMD Athlon processor 2.4GHz, 512 Mb RAM

A. PDB2PQR: to assign charges to atoms in pdbfile and optimize the hydrogen network using Amber (Charmm) forcefield. Finally this program creates a .pqr file for each subunit in a complex. Large scale PDB2PQR was automated by perl scripts.

B. APBS: to create .grd file for each .pqr file. Large scale APBS was automated by perl scripts.

C. ZDOCK: to combine two proteins together. Large scale ZDOCK was automated by perl scripts.

4.10 Routine Tasks:

MODELLER: For each sequence, ten models were generated by MODELLER, and the model with the lowest objective function score was selected as the representative model.

INSIGHT II binding site module and T-Coffee all use the default parameters.

When creating lipophilic potential surface with SYBYL, we chose "Gasteiger formal charges". "Create Surface" chose "Fast Connolly2" method. "Lipophilic Potential Opitions: Computation Mehtod" choose "protein". "Ramp Ranges" choose "User definied": Min value: -3.000, Max value: +1.5000. For other parameters we use the default values.

GRASP: Input Relative values: -5, 0, 5 to define the minumum, neutral and maxinum values for electrostatic potential map.

PDB2PQR: Version: 0.1.0; Force Field: Amber and Charmm. Perform the debumping operation; Perform hydrogen optimization; Perform hydrogen debumping; Perform water optimization.

APBS: Version: 0.3.1;

Parameters set in the Apbs.in file:

"read mol pqr pqrfilename end; elec name protein-name; mg-manual; dime 65 65 65; nlev 4; glen 96 96 96; gcent mol 1; mol 1; lpbe; bcfl sdh; pdie 2.0; sdie 78.54; ion 1 0.050 1.5; ion -1 0.050 1.5; pdie 1.0; chgm spl0; srfm smol; srad 0.0; swin 0.3; temp 298.15; gamma 0.105; write pot uhbd CDK6CGE2_DOCK_2 end; quit".

MolSurfer:

Each pdb file was split into two parts giving coordinates of the first and the second proteins forming the interface. These PDB files' derived .pqr files and .grd files work together as input file for MolSurfer.

3D-to-2D Interface Projection: MolSurfer defines interface to be clusters of points for which sum of the distances to the closest atoms of the two protein partners is less than 6 Å. Hetero-atoms that lie within 3 Å of any interface point were added to the 2D interface map. The properties of each protein were projected onto every point of the interface; these properties are assigned to the closest atom to the point.

Electrostatic Potential Calculation: Electrostatic potential of each (isolated) protein was computed from the .grd files generated with the APBS program which solved the finite difference linearized Poisson-Boltzmann equation. The potential values at each interface point were then interpolated by MolSurfer.

Residue Hydrophobicity: Residue hydrophobicities were assigned according to the residue name and following the parameters defined by Eisenberg D, *et al* (1982), namely:

ALA 0.25; GLN -0.69; LEU 0.53; SER -0.26; ARG -1.80; GLU -0.62;

LYS -1.10; THR -0.18; ASN -0.64; GLY 0.16; MET 0.26; TRP 0.37;

ASP -0.72; HIS -0.40; PHE 0.61; TYP 0.02; CYS 0.04; ILE 0.73; PRO -0.07; VAL 0.54;

If one belongs to none of the above, it will be set to zero.

Atom Hydrophobicity:

Atomic hydrophobicities were assigned according to the atom name and following the parameters defined by Eisenberg D et al (1989), namely:

'NZ LYS' -38; 'OE1 GLU' -37; 'C' 18; 'NH1 ARG' -38; 'OE2 GLU' -37; 'S' 5;'NH2 ARG' -38; 'OD1 ASP' -37; 'O' -9; 'OD2 ASP' -37; 'N' -9.

If one belongs to none of the above, it will be set as zero.

Atomic Radius:

Atomic radii were also assigned following those defined in Eisenberg D *et al* (1989): 'C', 1.9 Å; 'S', 1.8 Å; 'O', 1.4 Å; 'N', 1.7 Å. All the other atoms' radius was all assigned as 1.9Å.

ZDOCK: default options. The number of output prediction in each docking experiment is 2000.

Protein-Protein Contacting Residue Claculation:). If the distance of two residues coming from different chains of a complex structure is less than 6 Å, these two pairs are defined to be in contact. The calculation was done by perl script PDBcontactauto.pl (Appendix C). Perl script contactresiduepair.pl read in the output file of PDBcontactauto.pl and give a list the contacting residue pairs.

Protein-protein Interface Analysis Server: *http://www.biochem.ucl.ac.uk/bsm/PP/server*

All the paramaters of these web servers were set to default values.

The surface property parameters defined by PP server are listed below:

Accessible Surface Area (ASA): ASA is defined as the surface mapped out by the centre of a probe sphere, of radius 1.4Å, as if it were rolled around the van der Walss surface of the protein. For complexes, ASAs of each protomer in a complex and then the complete complex are calculated.

Planarity: The planarity of the interfaces is analysed by calculating the best fit plane through the 3-dimensional co-ordinates of the atoms in the interface using principal component analysis. The RMSD of the atoms on the plane is calculated and used as the measure of planarity.

Hydrogen bonds (HB): HBs are calculated by program HBPLUS, which generates a set of possible positions for a hydrogen (H) attached to a donor and then searches for donor (D) and acceptor (A) pairs that fit specified criteria as follows (where AA is the atom attached to the acceptor).

Distance Criteria	Angle Criteria
D-A < 3.9 Å	D-H-A > 90 degrees
H-A < 2.5 Å	D-A-AA > 90 degrees
	H-A-AA angle > 90 degrees

Table 4.2 Distance and Angle Criteria adopted by P-P Interface Analysis Server

Salt Bridges: Two oppositely charged atoms are defined to form salt bridge if they are less than or equal to 4.0Å apart.

Other web servers we used:

MAtDB: http://mips.gsf.de/proj/thal/db/

PDB: http://www.rcsb.org/pdb/

NPS@:http://npsa-pbil.ibcp.fr/cgibin/npsa_automat.pl?page=/NPSA/npsa_server.html

TIGR: http://www.arabidopsis.org/index.jsp

SAINT(DARWIN): http://raptor.scinq.org/~darwin/Saint3.html (default option)

TreeTop: http://www.genebee.msu.su/services/phtree_full.html (Algorithm: Cluster, Bootstrap required, for all the other options default value were chosen).

SUPERFAMILY: http://supfam.mrc-lmb.cam.ac.uk/SUPERFAMILY/. Default options.

PFAM: http://www.sanger.ac.uk/Software/Pfam/

SRS7: http://srs.ebi.ac.uk/srsbin/cgi-bin/wgetz?-page+srsq2+-noSession

HPRD: http://www.hprd.org/

Swiss-Prot: http://www.expasy.uniprot.org/database/knowledgebase.shtml

SAVS: http://nihserver.mbi.ucla.edu/SAVS/. This web server run following programs: ERRAT, Verify_3D, Prove, Pro_check, What_check, SFCHECK.

GeneDB: http://www.genedb.org/

5.

Results

5.1 Validation of Fundamental Premises

5.1.1 Modelling Control: Surface Property:

Mammalian cyclin-dependent kinases (CDK) have at least twelve subfamilies and cyclins have at least eleven subfamilies (Meyerson M, *et. al.*, 1999; Chen HH, *et al*, 2006; Grana X *et al*, 1994; Kikuno R, et al, 1999; de Graaf K *et al*, 2004). In plants, at least 6 subfamilies of CDKs and 7 subfamilies of cyclins have been found (Joubès J *et al*, 2000; Renaudin JP *et al*, 1996; Yamaguchi M *et al*, 2000; Menges M *et al*, 2005). Structures of only three human CDKs, three human cyclins, two viral cyclins, and three CDK-cyclin complexes are currently available in PDB (chapter 3). The high degree of sequence similarity (percentage sequence identity range between 35% and 75%) between CDK classes makes it possible that a CDK structure can be used as the modelling template (a CDK with known structure), for any target CDK (the CDK with unknown structure) of interest even if they belong to different subfamilies. The cyclin box, the first 5-helices repeat of the cyclin fold which is responsible for CDK-binding and activation, is conserved to a varying degree in all cyclin subfamilies. Although the cyclin super-family is quite diverse, the cyclin box region from one subfamily can also to be used as template for the target sequence come from a different subfamily (Brown NR, *et al*, 1995; Kim KK, *et al*, 1996).

Surface properties play a key role in protein-protein recognition. Whether it is possible at all for us to correctly predict the true CDK-cyclin pairings amongst modelled

structures of *Arabidopsis thaliana* CDK and cyclin homologues will depend on the answer to the following question: are the surface characteristics of the model structures more like their true structures than the template structures, even when templates come from different subfamilies? In this control experiment where the true structure is known, we modelled the human CDK2 using human CDK6 as template for comparative modelling. Modelling was carried out at several specific levels of misalignment in order to study the influence of misalignment on the surface properties of the model CDK2 structure. (i) Control1: The most reliable alignment, *i.e.* the best possible starting position: the alignment was prepared by superposing the target and the template structures. (ii) Control2a: structural alignment was used only in α-helix or β-strand regions and the pair-wise sequence alignment was edited manually in the remainder of the sequence using both sequence-similarity and knowledge of the template structure. (iii) Control2b: As control2a but only sequence similarity was used for editing the pair-wise alignment in non-secondary structure regions. (iiii) Control3; the worst starting position: the structure-derived pair-wise alignment by CE web server (Shindyalov IN, Bourne PE, 1998) was used only in α-helix and β-strand regions of CDK6 and all other regions were left for the modelling program to build automatically.

Comparison of electrostatic potential surface (drawn by GRASP (Nicholls *et al*, 1991)) between the observed structures of CDK2 and CDK6 and the modelled CDK2 structures showed that answer to the question is "yes". In Figure 5.1, column A shows the face CDKs present to cyclin partners and column B is the face revealed by a 180 degree turn about the y-axis of the structure. In column A all the interfaces are quite similar. Small

difference can be seen. For example in true CDK2 and the model CDK2s, most of the interfaces is blue but in CDK6 the blue region occupies the smallest area. In the faces of column B, it is clear that all the model CDK2 surfaces (except in Control 3) are more similar to the true CDK2 surface than to the template CDK6.

The Control2b case is representative of the comparative models generated in the study and used in our docking experiments. Generally we can say that comparative models' surface properties resemble their true structure surface properties closely enough to be probably useful for analyses considering surface properties.

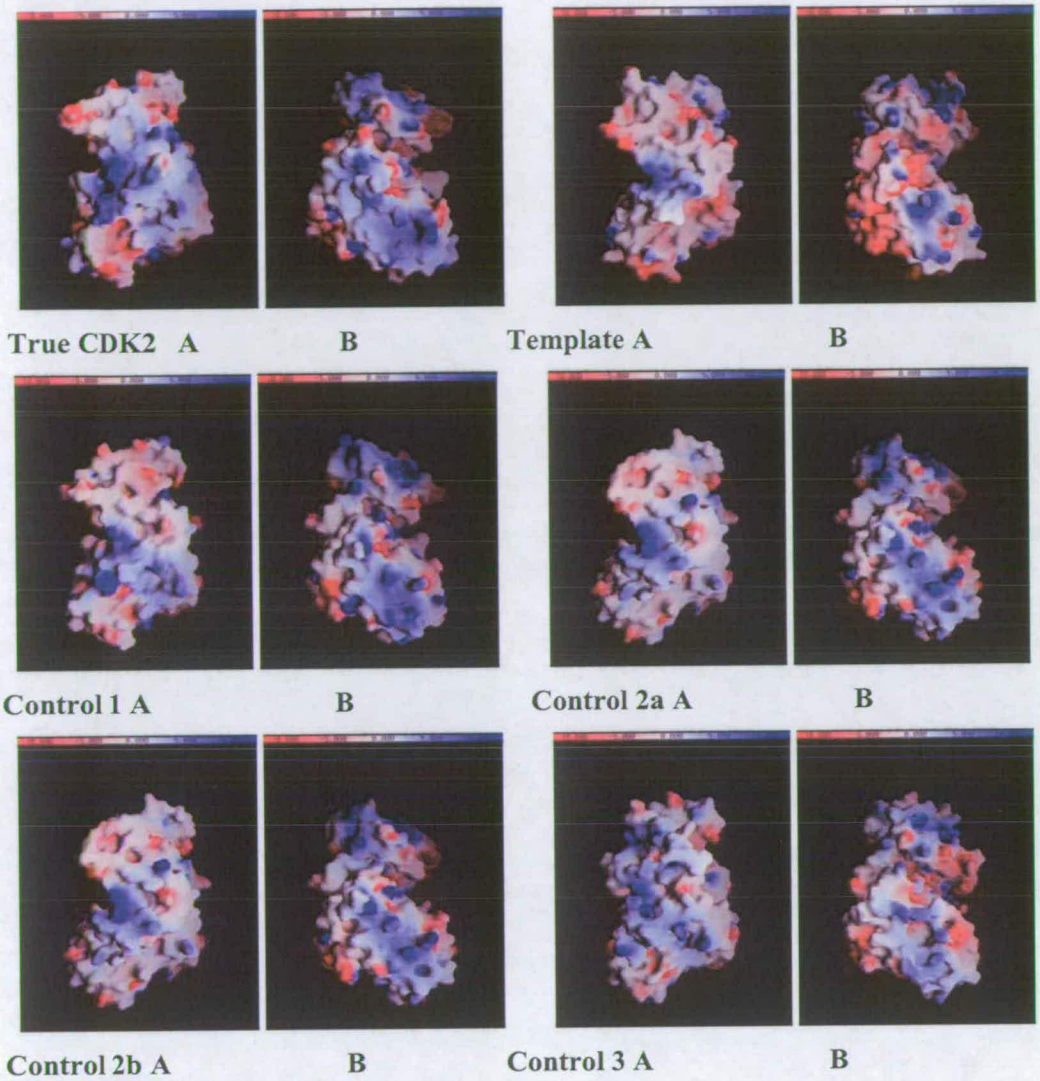


Figure 5.1. GRASP Electrostatic Potential Surfaces. Colour varies from -10 eV (red) to 0 eV (white) to 10 (blue). Column A is the face that CDKs present to cyclins. Column B is the face shown by a 180 degree turn about y-axis of the interface.

5.1.2 Conservation of CDK-cyclin binding site region

It is widely accepted that the binding site region of CDK-cyclin is conserved among different CDK-cyclin pairs and different species. This was confirmed by evolutionary trace analysis and structural comparison of known CDK-cyclin structures.

Evolutionary Trace Analysis In the case of the CDK family, 108 sequences which are composed of approximately 300 residues and come from a wide range of species, were identified and aligned to generate a sequence identity dendrogram. 30%, 40%, 50%, 60%, 70%, 80%, 90% pair-wise sequence identity percentages were used as partition identity cut-offs (PIC) to define different partitions of the dendrogram. Conserved residues and subfamily-specified residues are highlighted in different colours (refer to Figure5.2). On the interface with cyclin of the human CDK2 structure, both internal and external subfamily-specified and conserved residues are distributed inhomogeneously and form a single localized cluster. As expected, conserved traces are mainly located in the catalytic cleft of CDK2. In contrast, the CDK-cyclin binding sites are mainly subfamily-specific traces. This difference is easy to explain as CDK catalysis function is conserved within the entire CDK family, while CDK-cyclin binding specificity varies with different CDK subfamilies. At low PIC (30, 40 and 50), the conserved residues are always V18, K129, L133, which are external, and K33, R50, E51, D127, N132, G147, Y168, P177, R169, D185, which are internal residues (Lichtarge, 1996).

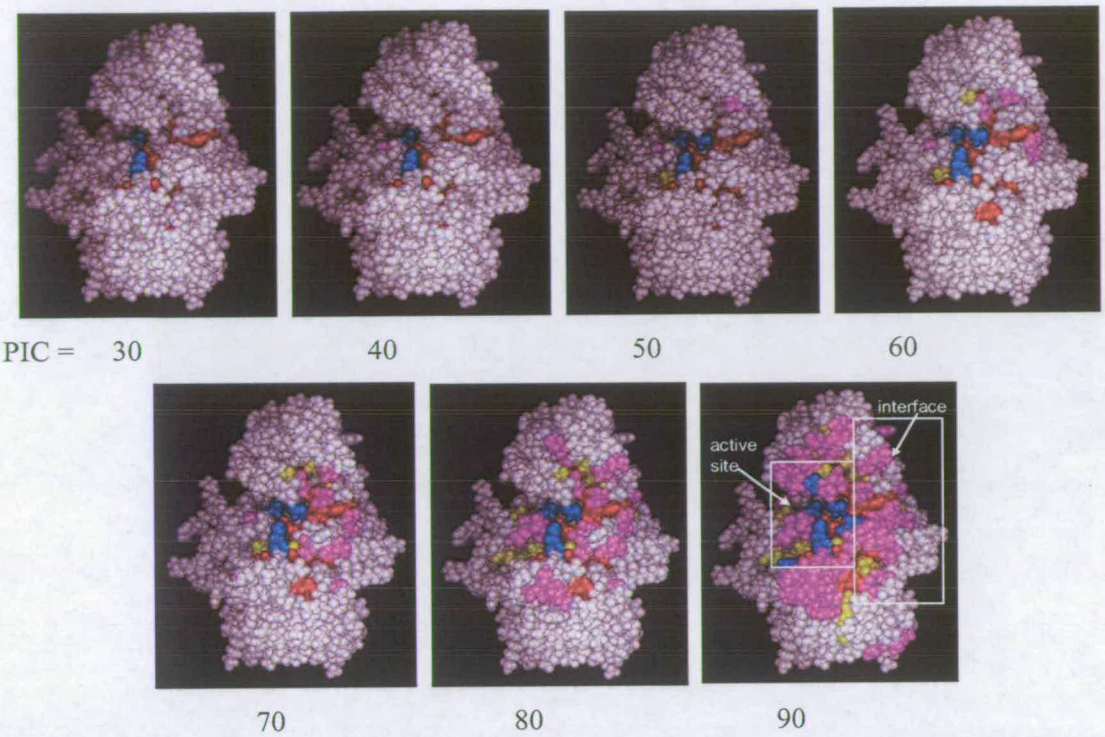


Fig.5.2 Evolutionary Trace of the complete family of CDKs mapped onto the 2.3 Å resolution structure of human CDK2. The internal positions that are conserved (red) and subfamily-specified (yellow), and the external subfamily-specified (purple) and conserved (dark blue) positions are distributed inhomogeneously and form a single localized cluster on this face (The face that contacts with cyclin, the active site and CDK-cyclin interface area are labelled out on the figure PIC = 90). This cluster locates mainly on the catalytic cleft of CDK2, with ATP and substrate binding sites mainly being conserved residues and the interface (between CDK and cyclin) mainly being subfamily-specific residues.

For the cyclin family, 98 sequences each of approximately 150 residues, were analysed (Fig 5.3). The structure is essentially free of trace signals at low PIC values (30, 40, and 50). Conserved positions are scarce in all the PIC partitions. One plausible reason for this is the high divergence of sequences between different cyclin subfamilies. Another putative explanation would be that cyclin-CDK binding specificity is not conserved within subfamilies.

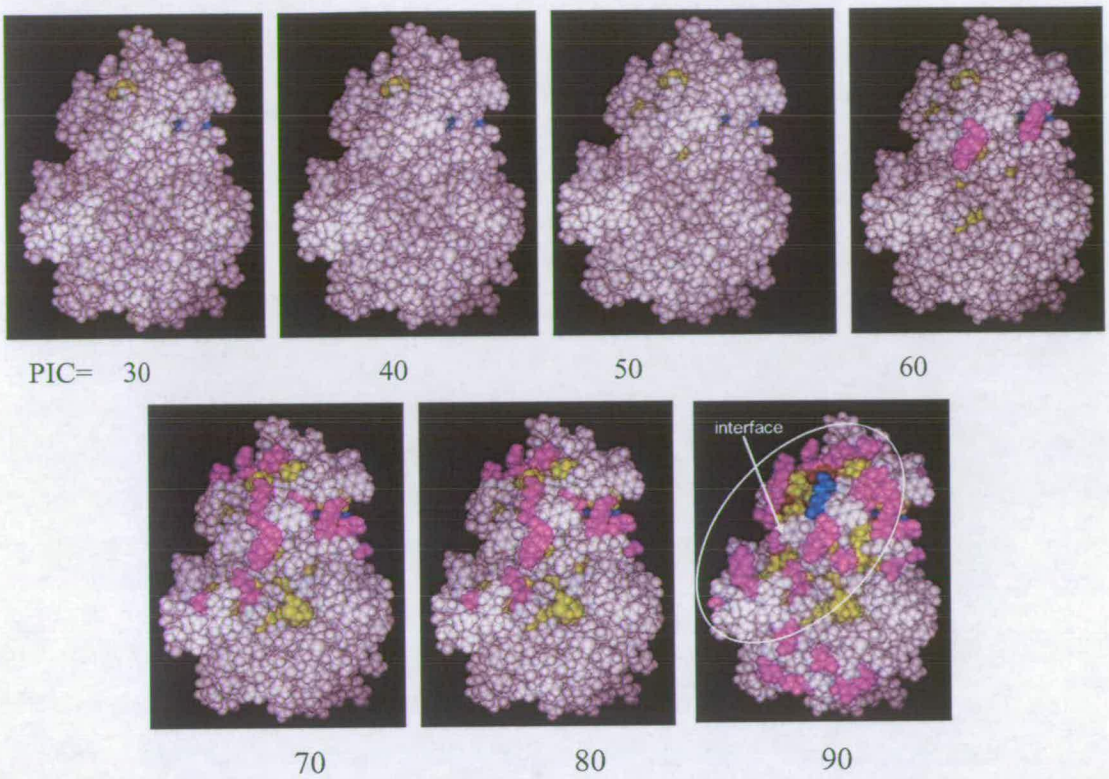


Fig.5.3 Evolutionary Trace of the complete family of cyclins mapped onto the 2.3 Å resolution structure of the human cyclinA2. This face (the face that contacts with CDK, interface area is labelled on the fig. PIC = 90) remains essentially free of trace signal until partition of PIC 60 (colour codes are the same as Fig 5.2).

Structural Comparison of Known Human CDK-cyclin Structures: In the superposed structures of the three known human CDK-cyclin complexes (the superposition was done with program QUANTA (Accdelrys Inc), all the three pairs interact roughly in the same region with similar orientation (Fig 5.4). RMSDs between their Cα atoms are lower than 15 Å (RMSD is the root mean square deviation of the positions of the backbone atoms, usually Cα atoms, between different proteins). The three CDKs were nearly perfectly superposed but cyclins not. This is because the superposition technique adopted by QUANTA is based on sequence alignment, and the fact that CDK sequences

are highly conserved but cyclin sequences are highly divergent. Generally, the binding sites on cyclins mainly involve the first cyclin fold and there is a large conservation of the residues located at the interface (Andersen G, *et al*, 1997). The RMSD between the Cα atoms of residues in CDK2 in contact with cyclin A and the Cα atoms of residues in CDK7 in contact with cyclin H is only 1.3 Å (Andersen built CDK7-cyclinH models based on CDK2-cyclinA2). The residues of cyclin H in contact with CDK7 are orientated in a conformation close to that in the complexed cyclin A. The main difference between residues of cyclin A and H that are in contact with CDKs are two loop residues (Glu117 and Phe118 for cyclin H and Glu269 and Ile270 in cyclin A). P25 and the first repeat of cyclin A also adopt a similar position on their cognate kinases except the N-terminal helix which is involved in binding with CDKs in cyclin A but not in P25 (Tarricone C, *et al*, 2001). The overall positional differences between P25 and cyclin A are that the loops between α1-α2 and α3-α4 of P25 approach CDK5 more closely than the equivalent segments of cyclin A.

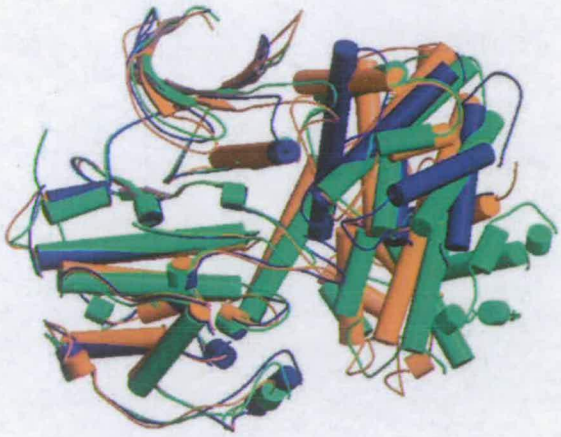


Figure 5.4. Subunit Orientation of three Crystal CDK-cyclin Structures: human CDK2-cyclinA2 (1FIN) (green), human CDK5-P25 (1H4L) (blue) and human CDK6-viral cyclin (1JOW) (brown). The diagram was created with MultiProt (Shatsky M., *et al*, 2002), MOLSCRIPT (Kraulis P.J., 1991) and RasMol (Milner-White & Sayle 1995).

5.2 Prediction Approach Development

5.2.1 *Arabidopsis* CDK/cyclin Structure Modelling

As only a few human CDK, cyclin or their complex structures are available in PDB, to predict the CDK-cyclin interactions, we need to model the structures of the 35 CDK-like sequences and 50 cyclin-like sequences by adopting comparative modelling.

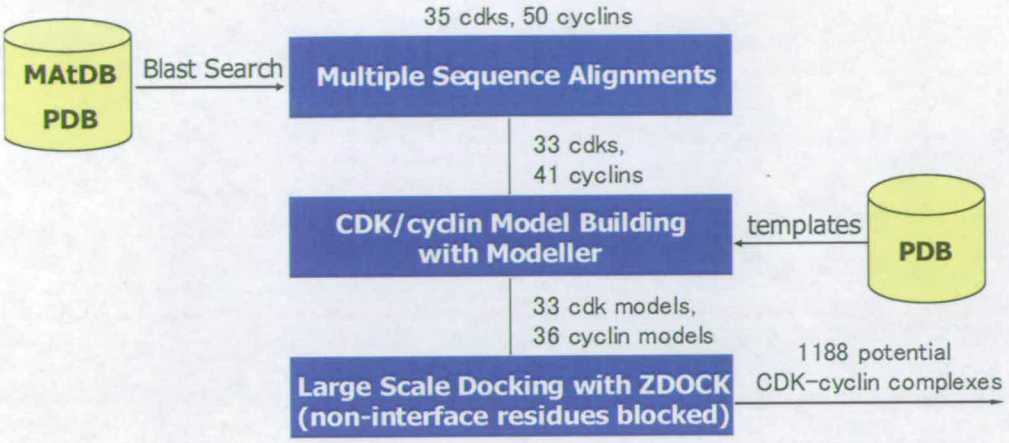


Figure5.5 Generation of potential *Arabidopsis thaliana* CDK-cyclin complexes through comparative modelling and large scale docking.

The comparative modelling strategy mainly consists of four procedures: template selection, alignment generation between target and template sequences, model building, and model evaluation. In our analysis, *Arabidopsis* and human CDK/cyclin homologue sequences were extracted from MAtDB and PDB using BLAST search (Fig 5.5). Expressions of some of these sequences have been experimental identified. Some sequences are only predicted genes (Table 5.1). Many of these genes have been annotated or named by Vandepoele and Menges (Vandepoele K, *et al*, 2002; Menges M, *et al*, 2005). Here we also allocate the sequences missed by Vandepoele and Menges to their sub-family based on the phylogenetic trees created by DARWIN server (Gonnet GH, *et al*, 2000). CDK-like sequences, AT1G10210, AT1G59580, AT4G36450, and AT2G18170, locate on the same sub-branch, and share the same changed characteristic "ALRTLRE" motif (discussed in more detail in Chapter 6, Figure 6.1 and Figure 6.2).

AT3G59790 locates on a neighbouring sub-branch close to the above four sequences and has an "AKRTLRE" motif which has only one, critical, residue change compared with the "ALRTLRE" motif. These five sequences are classified as putative mitogen-activated protein kinases. Among the cyclin-like sequences, some of them only have one cyclin structural repeat, but are not close homologues of P25 which is a human cyclin-like product having only the N-terminal cyclin structural repeat. These sequences form a distinct sub-branch on the phylogenetic tree and are named as CYCP family (Acosta JAT, *et al*, 2004).

Two cyclin-like sequences, Q38818 and Q9SGQ4, were extracted from the Swiss-Prot (Watanabe K and Harayama S, 2001) database. Q38818 locates on the sub-branch of CYCAs and can be treated as a close homologue of human cyclin A. Q9SGQ4 locates on the sub-branch of CYCBs and might be a close homologue of human cyclin B.

Gene Locus Name	Gene name*	Expression Status*	Gene Locus Name	Gene Name*	Expression Status*
AT3G48750	CDKA;1	Exp	AT2G44740	CYCP4;1	Pre
AT1G18040	CDKD;3	Exp	AT2G45080	CYCP3;1	Pre
AT1G66750	CDKD;2	Exp	AT3G21870	CYCP2;1	Pre
AT1G73690	CDKD;1	Exp	AT3G60550	CYCP3;2	Pre
AT1G03740	CKL6	Exp	AT3G63120	CYCP1;1	Exp
AT1G09600	CDL11	Exp	AT5G07450	CYCP4;3	Pre
AT1G10210	MPK1	Pre	AT5G61650	CYCP4;2	Pre
AT1G18670	CKL3	Exp	AT1G15570	CYCA2;3	Exp
AT1G20930	CDKB2;2	Exp	AT1G16330	CYCB3;1	Exp
AT1G33770	CKL14	Exp	AT1G20610	CYCB2;3	Exp
AT1G53050	CKL15	Exp	AT1G44110	CYCA1;1	Exp
AT1G54610	CKL9	Exp	AT1G47210	CYCA3;2	Exp
AT1G57700	CKL10	Exp	AT1G47220	CYCA3;3	Exp
AT1G59580	MPK2	Pre	AT1G47230	CYCA3;4	Exp
AT1G67580	CDKG;2	Exp	AT1G76310	CYCB2;4	Exp
AT1G71530	CKL12	Pre	AT1G77390	CYCA1;2	Exp
AT1G74330	CKL2	Exp	AT1G80370	CYCA2;4	Pre
AT1G76540	CDKB2;1	Exp	AT2G17620	CYCB2;1	Exp
AT2G18170	MPK7	Pre	AT2G26760	CYCB1;4	Exp
AT2G38620	CDKB1;2	Exp	AT3G11520	CYCB1;3	Exp
AT3G05050	CKL8	Exp	AT4G35620	CYCB2;2	Exp
AT3G54180	CDKB1;1	Exp	AT4G37490	CYCB1;1	Exp
AT3G59790	MPK10	Pre	AT5G06150	CYCB1;2	Exp
AT4G10010	CKL13	Pre	AT5G11300	CYCA2;2	Exp
AT4G22940	CKL4	Pre	AT5G25380	CYCA2;1	Exp
AT4G36450	MPK14	Pre	AT5G43080	CYCA3;1	Exp
AT5G10270	CDKC;1	Exp	Q38818	Cyclin; 2	Exp
AT5G39420	CKL1	Pre	AT1G70210	CYCD1;1	Exp
AT5G44290	CKL5	Exp	AT2G22490	CYCD2;1	Exp
AT5G50860	CKL7	Exp	AT3G50070	CYCD3;3	Exp
AT5G63370	CDKG;1	Exp	AT4G03270	CYCD6;1	Exp
AT5G63610	CDKE;1	Exp	AT4G34160	CYCD3;1	Exp
AT5G64960	CDKC;2	Exp	AT4G37630	CYCD5;1	Exp
AT4G28980	CDKF;1	Exp	AT5G02110	CYCD7;1	Exp
AT1G27630	CYCT;1	Exp	AT5G65420	CYCD4;1	Exp
AT4G19600	-	Pre	AT5G67260	CYCD3;2	Exp
AT4G19560	-	Pre	Q9SGQ4	Cyclin;2	Exp
AT5G48640	CYCC1;1	Exp	AT5G10440	CYCD4;2	Exp
AT5G48630	CYCC1;2	Exp	AT1G14750	SDS	Pre
AT5G27620	CYCH;1	Exp	AT1G20590	CYCB2;5	Exp
AT1G34450	-	Pre	AT5G45190	-	Pre
AT5G27620	-	Pre	AT2G26430	CYCL1	Exp
AT1G35440	-	Pre			

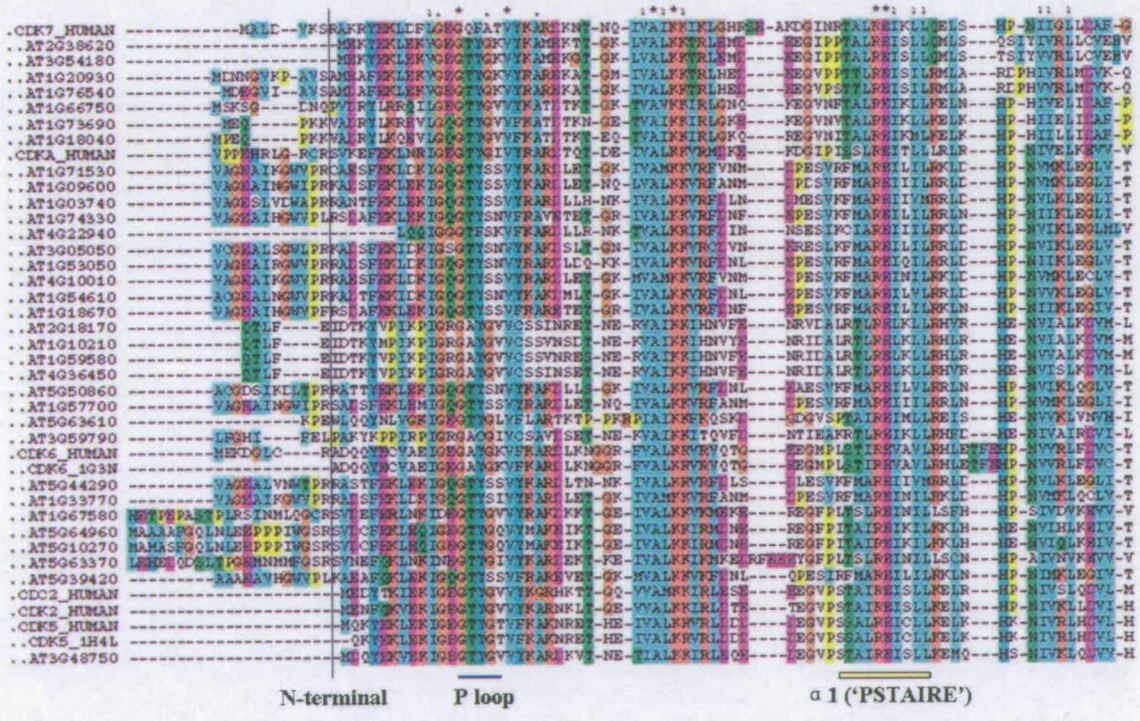
(to be continued)

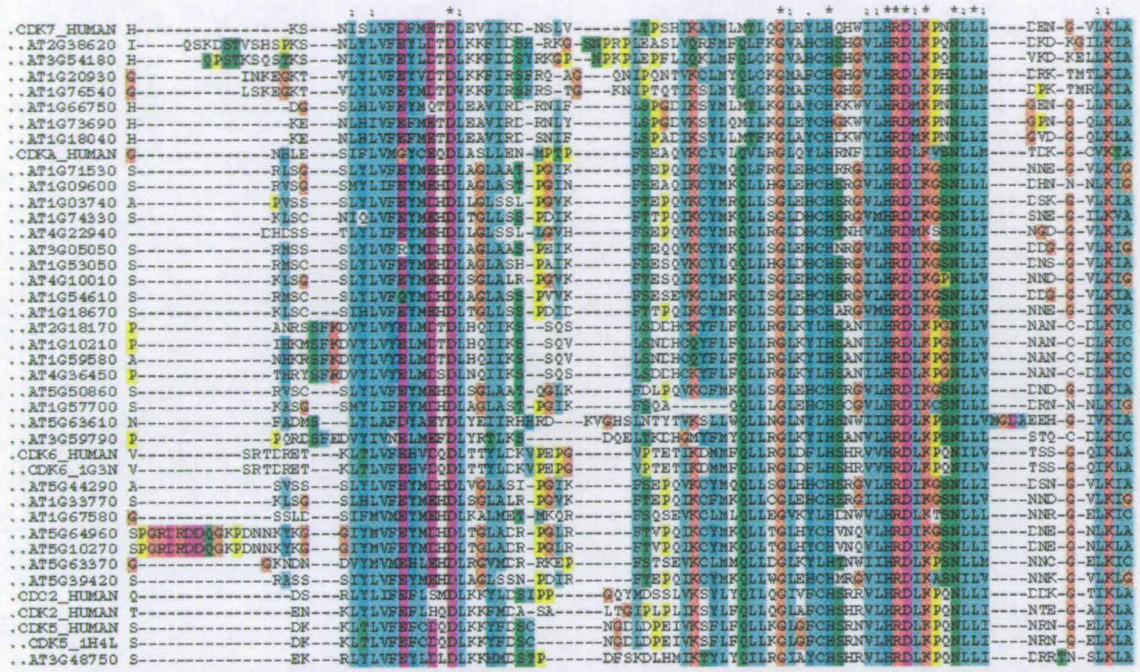
Table 5.1 List of all 85 core cell cycle gene in *Arabidopsis thaliana*. Gene IDs highlighted in red color were finally abandoned either due to the absence of a suitable template, or to alignment difficulty.

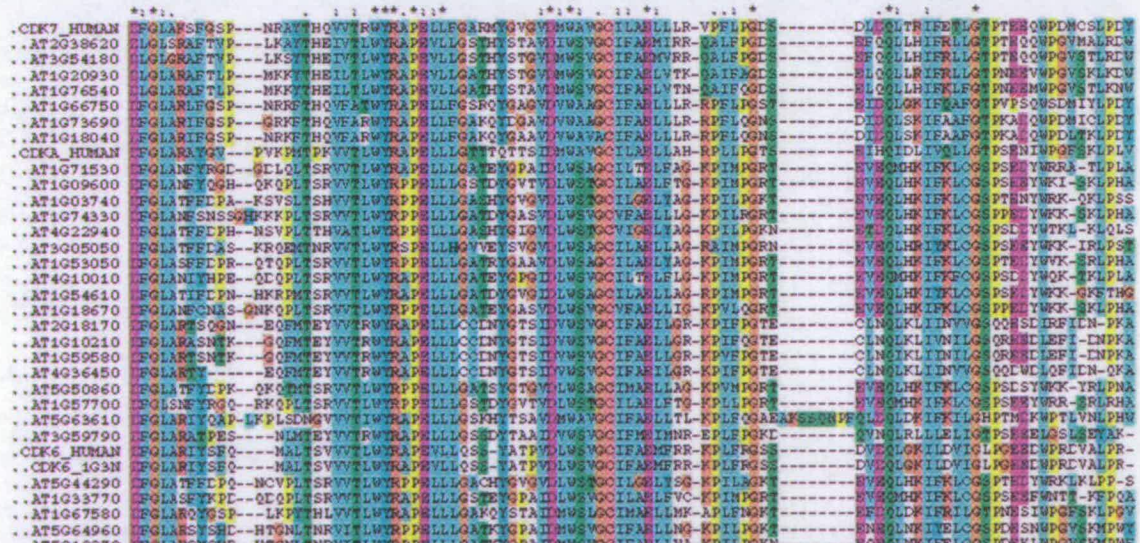
* Vandepoele K, *et al*, 2002; Menges M, *et al*, 2005, Swiss-Prot Database; SIGNAL database (Yamda K, et al, 2003); Acosta JAT, *et a*l, 2004.

Multiple alignments of these sequences were either built using program T-Coffee (Notredame *et al*, 2000), SUPERFAMILY HMMs (Gough *et al*, 2001), or directly extracted from PFAM (Bateman *et al.*, 2004) (Figure 5.6, 5.7, 5.8) and manually edited. Some *Arabidopsis* cyclin-like sequences contain only the cyclin box instead of the full cyclin core (Figure 5.8). The C-terminal residues of these sequences were poorly aligned. However this should not affect the model qualities of the N-terminal cyclin box which is involved in the CDK-cyclin interactions. Model building and evaluation were carried out with the program Modeller6v2 (Sali & Blundell, 1993). Alignment generation is the most important procedure in comparative modelling and has great impact on the quality of final models. Usually, when the sequence identity percentage (SIP) between target and template is higher than 40%, the model should be correct (MartÃ -Renom MA *et al*, 2002). Here we need to bear in mind that this SIP number comes from pair-wise alignment and is usually higher than SIPs calculated from multiple sequence alignments though multiple sequence alignment is much more reliable than pair-wise alignment. SIPs calculated in our work come from multiple alignments that were used later to build models. The sequence percentage identities between human and *Arabidopsis* CDK sequences are between 35%-70% and it was not difficult to create reliable alignments.

Cyclins are much more diverse in sequence but highly conserved in structure (Fig 5.7, 5.8). The powerful tools, the SUPERFAMILY HMMs, were used to build the alignment between human and *Arabidopsis* cyclin homologue sequences. However, some sequences still cannot be extensively aligned with template sequences without opening too many gaps, probably due to gene prediction errors and their very low sequence similarities. No suitable template structure is available for some sequences; for example, the CDK7 homologue sequences in *Arabidopsis*. All these sequences were abandoned to improve the overall quality of model structures. Templates for each target sequence were selected based on PAM distances on phylogenetic trees created by the DARWIN server.






αL12
T-loop

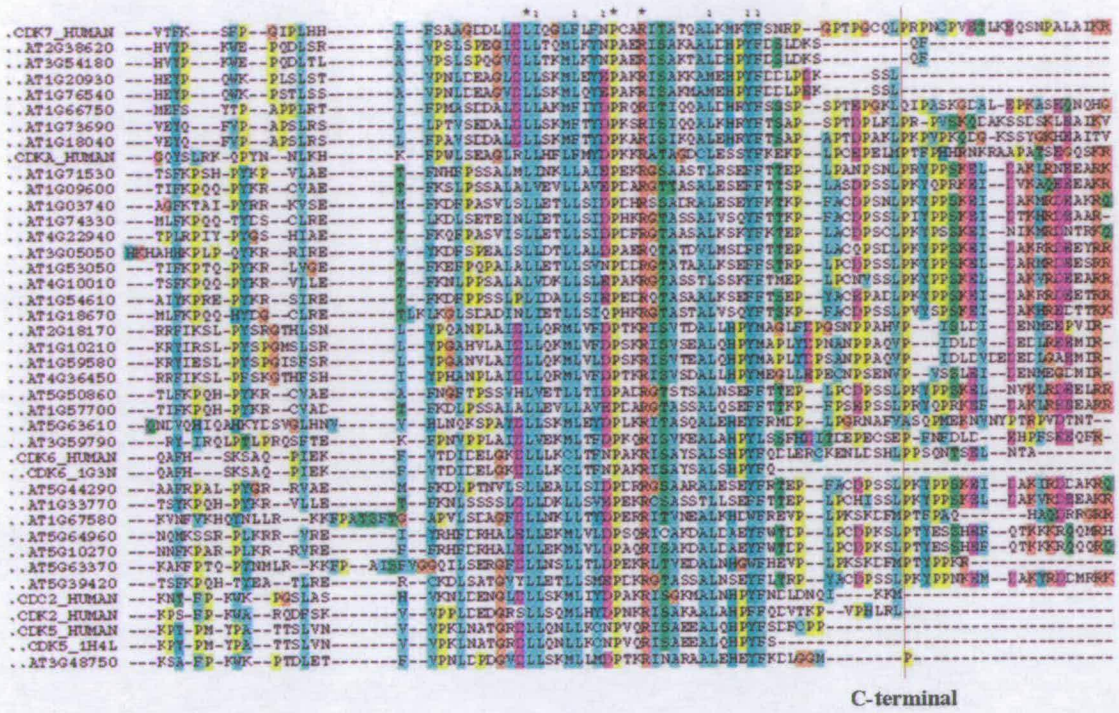


Figure5.6 Multiple sequence alignment of *Arabidopsis* and human CDK-like sequences. The alignment was generated by T-Coffee. Positions of P loop, helix α1, helix αL12, T loop (activation loop) of human CDK2 are labelled. The starting and ending positions for comparative modelling are indicated (as N- and C- terminal). Colouring scheme of ClustalX is applied and described in the Appendix B.

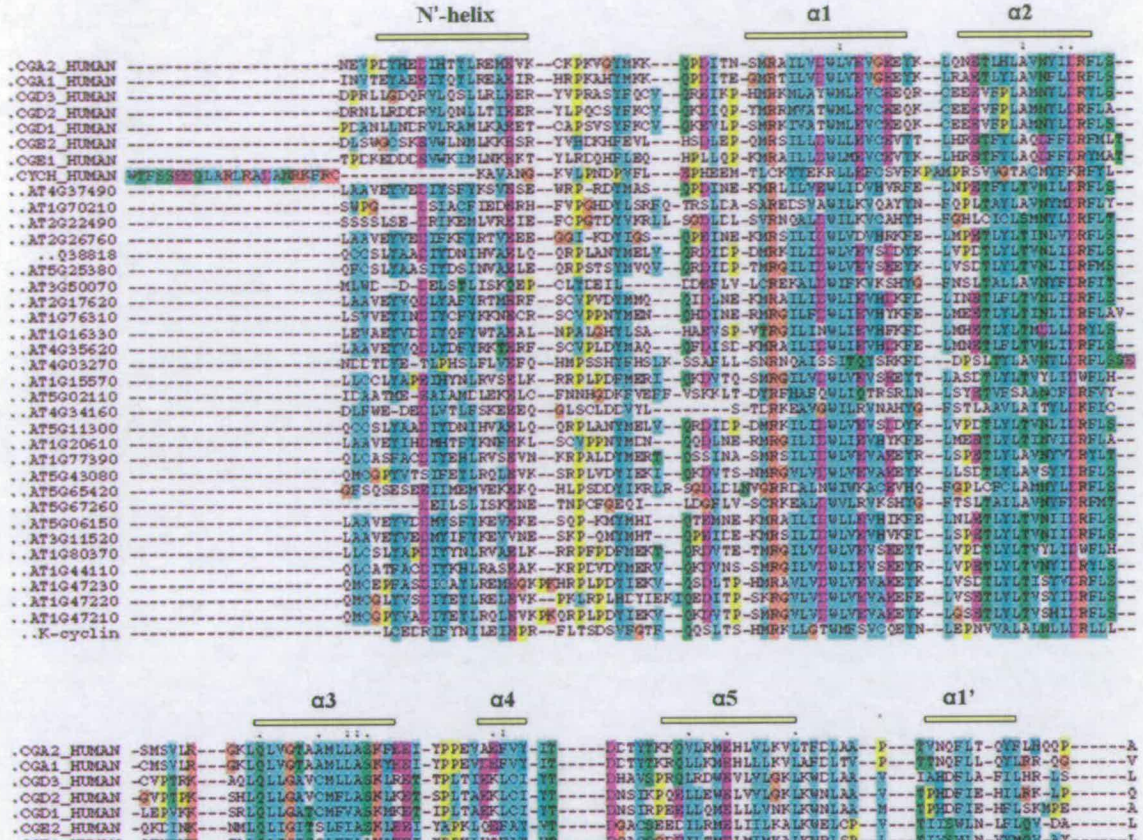

N'-helix
α1
α2
α3
α4
α5
α1'
.CGA2_HUMAN
.CGA1_HUMAN
.CGD3_HUMAN
.CGD2_HUMAN
.CGD1_HUMAN
.CGE2_HUMAN
.CGE1_HUMAN
.CYCH_HUMAN
..AT4G37490
..AT1G70210
..AT2G22490
..AT2G26760
..Q38818
..AT5G25380
..AT5G02110
..AT1G76310
..AT1G16330
..AT4G35620
..AT4G03270
..AT1G15570
..AT5G02110
..AT4G34160
..AT5G11300
..AT1G20610
..AT1G77390
..AT5G43080
..AT5G65420
..AT5G67260
..AT5G06150
..AT3G11520
..AT1G80370
..AT1G44110
..AT1G47230
..AT1G47220
..AT1G47210
..K-cyclin

Figure 5.7 Multiple sequence alignment of *Arabidopsis* and human cyclin-like sequences containing the full structural cyclin core. The alignment was built by SUPERFAMILY HMM. The positions of the N terminal helix, helices α1-5, and helix α1' of human cyclin A2 are labelled. The colouring scheme of ClustalX is applied.

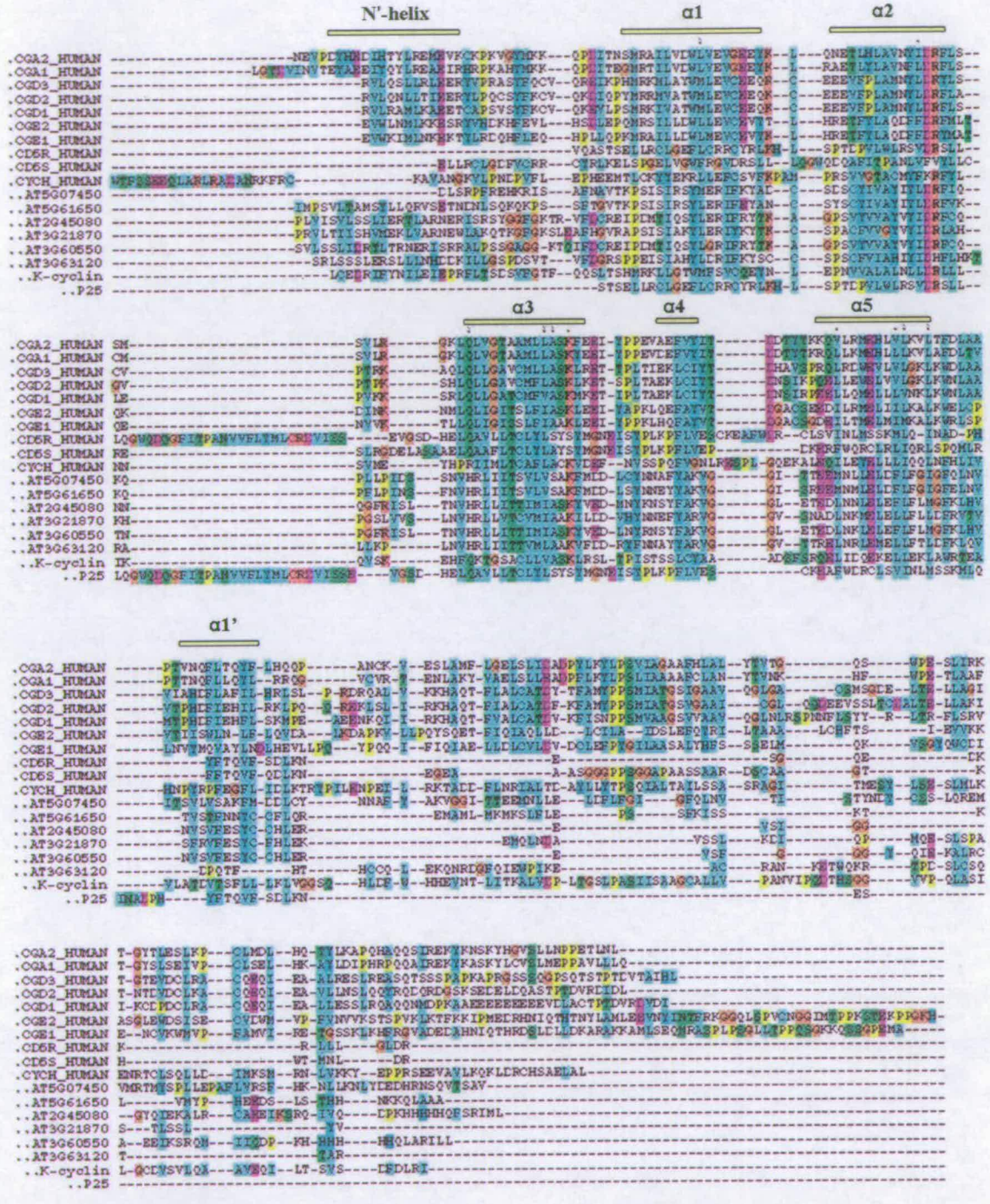
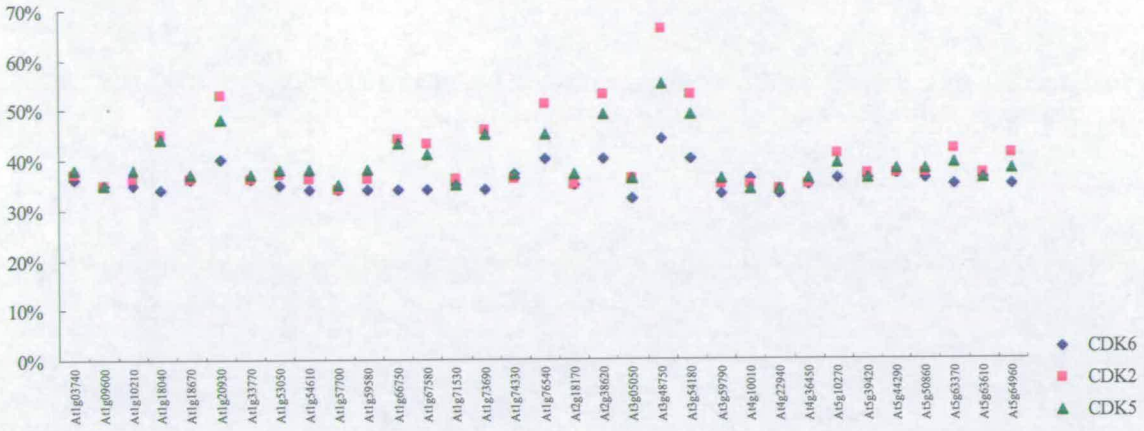
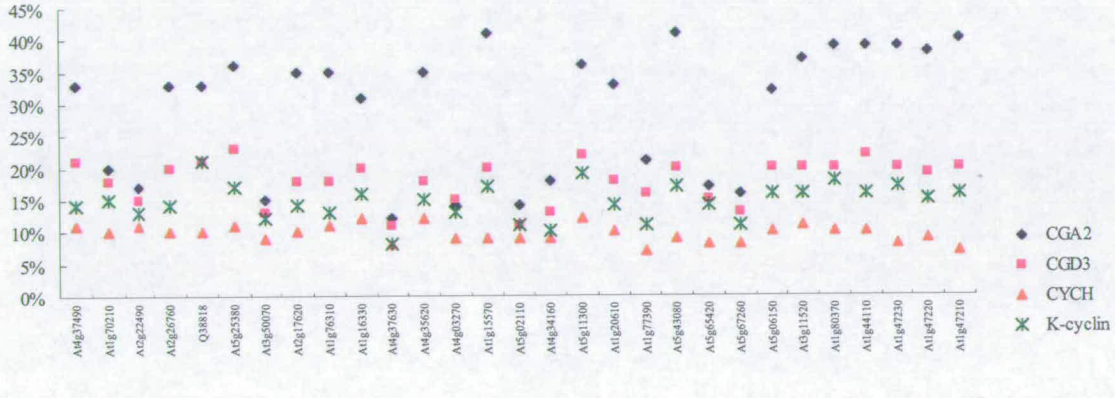


Figure 5.8 Multiple sequence alignment of human cyclin sequences and *Arabidopsis* cyclin-like sequences containing only the cyclin box. The alignment was built by SUPERFAMILY HMM. The positions of the N-terminal helix, helices α1-5, and helix α1' of human cyclin A2 are labelled. The colouring scheme of ClustalX is applied.

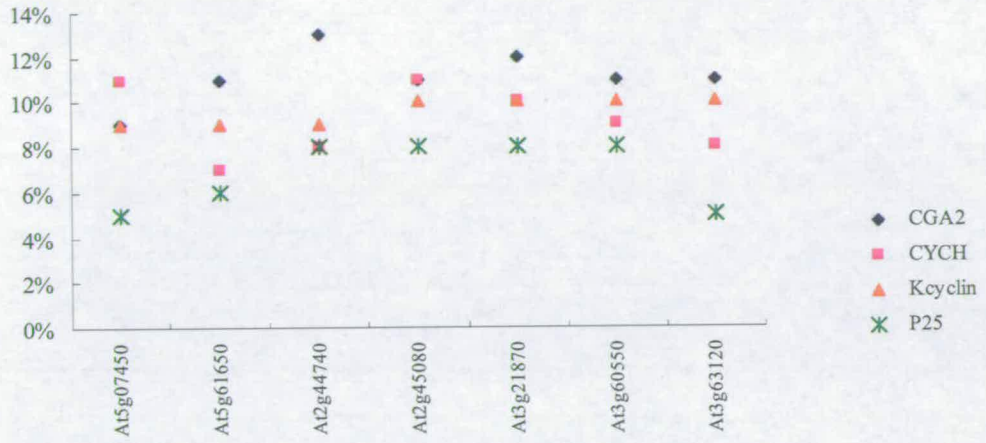


A



B

(to be continued)



C

Figure 5.9 Sequence percentage identities between *Arabidopsis* target sequence and template (human CDK/cyclin). A: CDK homologues. B and C: cyclin homologues. Sequence percentage identities were calculated based on the multiple alignments used for their model building.

The DARWIN server builds the phylogenetic tree of proteins directly based on their unaligned protein sequences. This is a special advantage in the cases when satisfactory alignments of sequences of an entire protein family (subfamily) are difficult to build. However, the DARWIN server lacks the bootstrap procedure which helps to estimate the stability of the trees. So we also built phylogenetic trees of these sequences using the TreeTop web server (http://www.genebee.msu.su/services/phtree_full.html) with bootstrap values based on the alignments we have created. Generally, all the sequences' locations on the sub-branches of the DARWIN trees and TreeTop trees are consistent (Chapter 6 figure 6.1 and 6.2).

Among the four steps of our comparative modeling strategy, two steps need to be repeated frequently to get the most recent data. One procedure is the first step, extracting the *Arabidopsis* CDK/cyclin like sequences. These protein sequences come from gene prediction based on EST/cDNA data. These sequences were frequently updated to minimize the gene prediction error and build new comparative models with the latest sequences before finally using the models to run docking experiments. The sequences extracted from MAtDB during our latest updating (Aug.22.2005) were compared with the "old" sequences finally used for model building and docking. Most of the CDK-like sequences had not changed during this time period. However, the C-terminal residues of AT2g38620 are different from the previous version. This change is not compatible with the cDNA data stored in the SIGnAL database (http://signal.salk.edu/). The cDNA data is consistent with the sequence we used for modeling. This sequence looks more plausible than the new version in a multiple sequence alignment of *Arabidopsis* CDK-like sequences. The C-terminal region was more fully aligned with other CDK-like sequences. This C-terminal change located far from the interface region and should not affect its structure. For cyclin-like sequences, there is a long 14 amino acid deletion in the new AT3g11520 sequence compared with the old sequence. But this deletion is not located on the cyclin core part and did not have any impact on our modeling of this protein structure.

Another step which needs frequent updating is the extraction of templates from PDB. We need to check whether there are new template structures that have become available. No new template structure was stored in PDB to the date this project was finished.

In our comparative modelling, template structures of CDKs which undergo conformational change during association with cyclin were all extracted from their complexed structures with cyclins. Cyclins undergo no significant conformation change during association with CDK. Structures coming from the cyclin chain in a complex and monomeric cyclin, can both be used as template for modelling (Jeffrey P.D. et al., 1995). In this way the problem of loop motion at the protein-protein interface was reduced to local side-chain motions. Consequently, we can use rigid body docking programs that are able to tolerate considerable structural deformation and atomic clashes (CAPRI result, Janin J, 2005).

The protein side-chain modelling program, SCWRL (Canutescu AA, *et al*, 2003), can generally improve the side-chain quality of models. We carried out a set of control experiments by analyzing twenty transient protein complexes with known structures. Model structures were built by MODELLER which directly modeled the complexes onto their crystal structures. Then improved models were made by modifying the side-chain part of models made by MODELLER with SCWRL. All these complex structures were split into two separate chains and recombined with two different docking programs: ZDOCK and ClusPro (Comeau *et al*, 2004). In our investigation to combine human CDK2 and cyclin A with ZDOCK and ClusPro, ZDOCK gave much higher ZDOCK scores for crystal structures of CDK2/cyclinA, and models built only with MODELLER than the models with side-chains refined by SCWRL (table 5.2). The complex structures made by docking programs with source structures coming from crystal structures and MODELLER models are also more correct than the complexes with source structures

coming from SCWRL refined models. Generally we can say that docking programs like the side-chain geometry of crystal protein structures and the Modeller version models, but not the side-chain coordinates built by SCWRL. We got similar results In the investigation of the other 19 transient protein-protein complexes. A plausible reason is that SCWRL puts side chains in their minimum energy position and for side chains on the surface this is often pointing straight out into solution. This would mean a SCWRL prediction has a lot of little spikes, for example Arg and Lys, pointing outwards and is not easy to dock together. Because Modeller has an electrostatic energy function, some of a model's spiky side-chains may end up folding back onto the protein to make hydrogen bonds. This does not mean they are correct predictions, but if sidechains are not sticking out protein's surfaces they may be easier to dock to (personal communication from Roland L Dunbrack Jr.).

Source Structures	ZDOCK Score	Cα RMSD (ZDOCK) (Å)	Cα RMSD (ClusPro) (Å)
Crystal	106.36	0.80	1.26
MODELLER	106.31	0.79	1.35
SCWRL	57.71	1.69	Wrong orientation*

Table 5.2 The features of docked complexes of human CDK2-cyclinA with different source structures: crystal, models built with MODELLER, models built with MODELLER and refined with SCWRL. Cα RMSD between docked complexes and crystal CDK2-cyclin A complex (1FIN) were calculated by program QUANTA (Accelry Ltd.). *- In the docked complex generated by ClusPro CDK2 and cyclin A, subunits were combined in a completely wrong relative orientation.

Sequence percentage identity can be used as indications of model quality (MartÃ - Renom MA *et al*, 2002). As can be seen from Figure 5.6, all the *A. thaliana* cdk-like sequences have high percentage sequence identity to the template sequences with range from 35% to 75%. If these SPIs were calculated based on pair-wise alignments, they usually became even higher. Some *A. thaliana* cyclin homologues had templates with high sequence identity, that is, larger than 30%, and their models generally would be correct. The other cyclin models need further inspection.

Several programs that are designed for model quality evaluation were used here to estimate the quality of different aspects of these models. These programs include Procheck, Verify_3D, ERRAT and Prove.

The percentage of residues in core and allowed regions of the Ramachandra plots calculated by Procheck evaluate the stereochemical quality of models. The percentage of residues where backbone angles fall in the most favoured regions plus that in additional allowed regions is an important index for the model quality. If this percentage is above 90%, the model quality is good and there are not many stereo-chemical clashes. The percentages of all our models are above this threshold (Figure 5.10 A). This is mainly because we used Modeller to build our models and Modeller is very good at avoiding stereo-chemical clashes as it uses a force-field (Charmm) to carry out energy minimization during model building. Similar results were obtained in the evaluation using PROVE. In this method, the percentage of buried "outliers" is calculated. The "outliers" are defined in Prove to be the structures with volume-z score exceeding their limits. The limits were derived from the Z score RMSD distribution (Pontius J *et al.*, 1996). If the proportion is less than 1%, the model quality is satisfactory

(http://nihserver.mbi.ucla.edu/SAVS/). This criterion was met by all our models. Verify-3D calculated the statistical preferences, viewed as 3D-1D scores of each of the 20 amino acids for the environment of each residue position in the 3D model. Most of our models got satisfactory or allowed values in Verify_3D evaluation (Figure 5.10 B). The models with 3D-1D percentage lower than 55%, AT2g44740, AT5g07450, and AT4g37630, need to be carefully inspected in the final prediction result. ERRAT gives the percentage of the sequence that is above the 95% confidence limits for each chain (http://nihserver.mbi.ucla.edu/SAVS/). Most of the model percentages are over 60% (Figure 5.10 C) and are acceptable. The models with ERRAT percentage lower than 60%, AT1g18040, AT1g66750, AT1g73690, AT5g50860, will need inspection in our final prediction result.

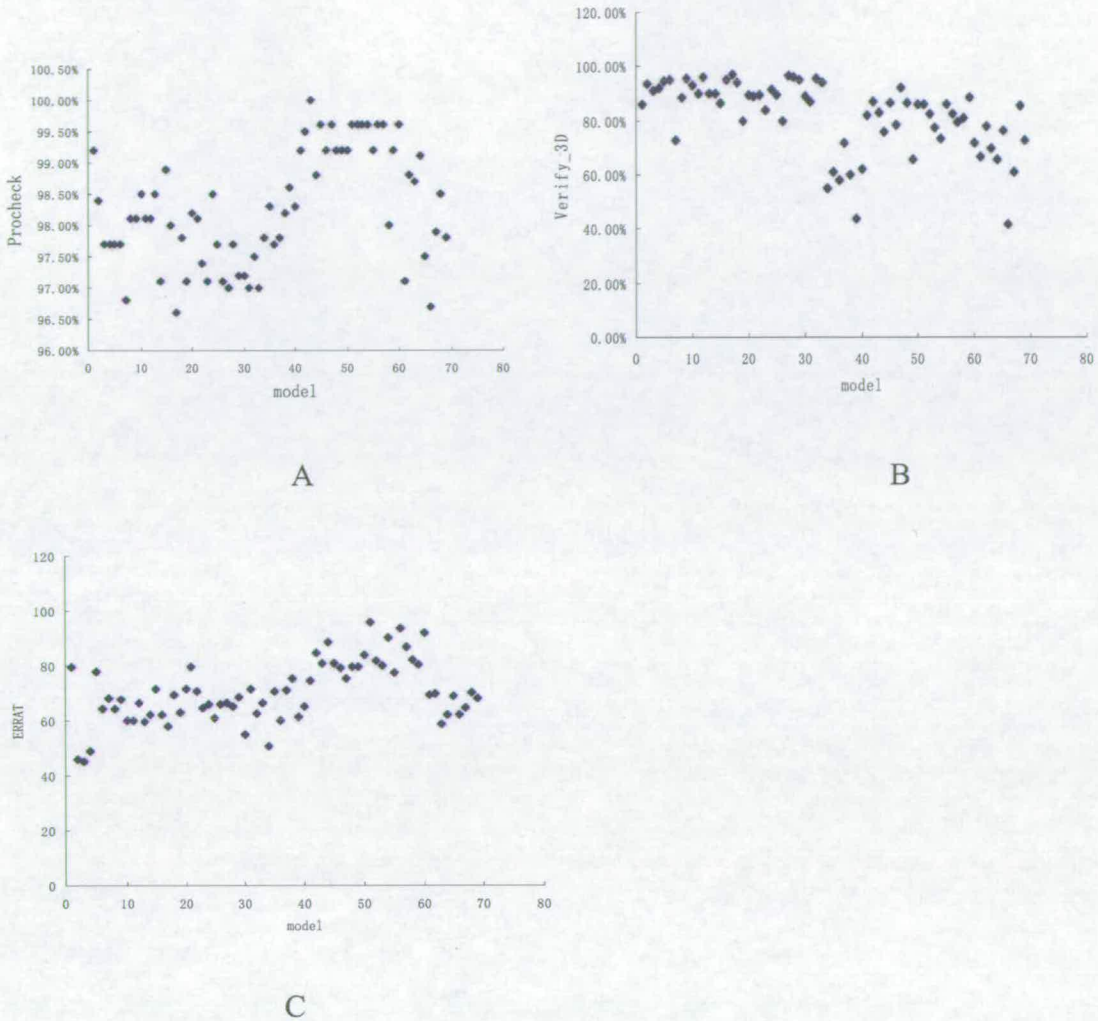


Figure 5.10 Plots of evaluation values for all the *Arabidopsis* cdk/cyclin models using different evaluation methods: Procheck (A), Verify_3D (B) and ERRAT (C).

5.2.2 Large Scale Docking Approach

Based on the fact that the binding site region of the CDK-cyclin complex is conserved in different CDK-cyclin pairs and in different species, we blocked the residues which we sure are not on the interface so that the dock program only searches different orientation and translations of interacting site regions. Ideally, this process should be able to greatly

reduce the running time of each docking experiment and increase the prediction accuracy of the dock program, since most of the highest dock score usually represents the correct subunit orientation of the docking pairs. The interface residues of model structures came from interface residues on their corresponding template structures by extracting target residues in the same position as template interface residues and the flanking four residues of the alignment (Figure 5.11).

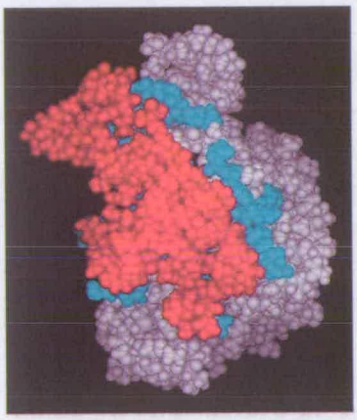


Figure 5.11 An example of blocking non-interface residues on proteins before docking. Residues of human CDK2 whose side-chain atoms are in 5Å distance with side-chain atoms of human cyclin A2 are highlighted in red. Residues highlighted in cyan are neighbouring residues of interface residues in sequence. These neighbouring residues are not blocked in order to give docking program a certain searching freedom.

ZDOCK (Chen R *et al*, 2003) was selected from many protein-protein docking programs in our prediction approach mainly because it pays a lot of attention to electrostatic interactions. As we use model structures to make predictions, programs that mainly work on shape complementarity might not be suitable tools. At the same time, ZDOCK

itself is also an outstanding protein-protein docking program with relatively high accuracy.

The impact of masking non-interface residues on docking was investigated. Here we also test the ZDOCK prediction accuracy in the case of interacting region being unmasked: human CDK2-cyclinA and CDK5-P25 crystal structures, *Xenopus laevis* and *Carassius auratus* CDK2-cyclinA2 model structures are split into separate chains, then these pairs were recombined together using ZDOCK. Generally, the running time for each dock experiment is reduced from 8-12 hours in unmasked cases to 3-5 hours in masked cases. The ZDOCK scores and associated re-docked complex structures remain nearly the same in these unmasked cases as in the masked cases. However, for several model structures of *Arabidopsis,* in unmasked cases the highest scores no longer refer to the correct interacting region and orientation, neither are they the same nor similar to the highest ZDOCK scores in masked cases. Therefore masking non-interface residues does help improving the prediction accuracy of ZDOCK.

The accuracy and reproducibility of ZDOCK with non-interface residues masked was carefully studied to use as positive control experiments. Separate chains in the crystal structures CDK2-cyclinA, CDK5-P25, model structures CDK6-cyclinD1, and CDK2-cyclinA of *Xenopus* were recombined together by ZDOCK repeatedly with each chain original orientation changed in every round. The highest ZDOCK scores and the associated docked complex structures remain nearly the same for each pair in all the repeats. Cα atom RMSDs calculated by QUANTA between docked complex structures associated with the highest ZDOCK scores and the corresponding crystal or model complex structures are 0.8Å for human CDK2-cyclinA2, 0.54Å for human CDK5-P25,

1.02Å for human CDK6-cyclin D1 and 0.91Å for *Xenopus* CDK2-cyclin A2 (Figure 5.12).

Figure 5.12. Superposed structures in positive control experiment. The crystal structure of human CDK2-cyclin A2 is coloured in blue. The positive control structures (docked structures of human and *Xenopus* CDK2-cyclinA2) are highlighted in green and yellow respectively. The docked structures of Pair I and Pair II are highlighted in cyan and brown respectively.
The diagram was created with MultiProt (Shatsky M., *et al*, 2002), MOLSCRIPT (Kraulis P.J., 1991) and RasMol (Sayle RA & Milner-White EJ, 1995).

In our prediction approach, non-interface residues on the 33 *Arabidopsis* CDK and 36 cyclin models were masked and then ZDOCK was run for all these CDK and cyclin pairs (Fig 5.12, 5.13). After about four mouths, we got the ZDOCK scores and corresponding complex structures of 1188 pairs. A set of additional selection criteria was applied further to extract the most likely interacting pairs from these pairs.

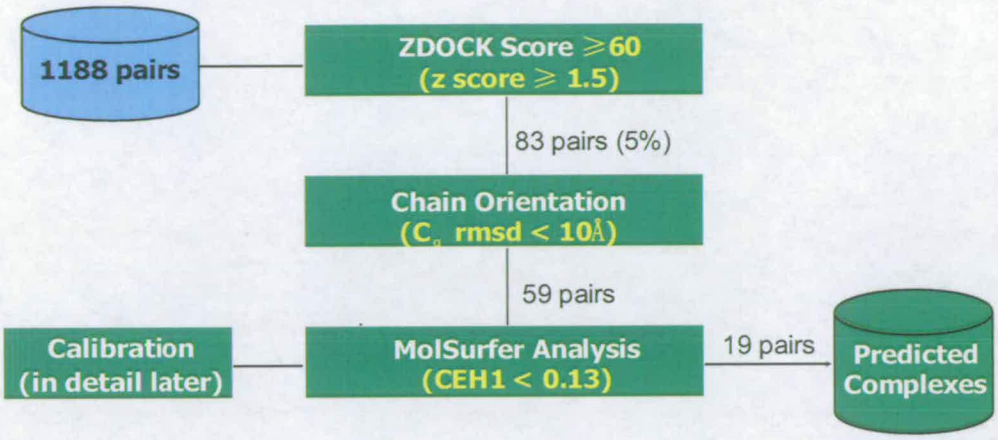


Figure 5.13 Flowchart of our partner prediction procedures.

5.2.3 Selection Criterion Calibration

5.2.3.1 ZDOCK Score and z score

The docking yielded 1188 possible interacting pairs. Most of these pairs' highest ZDOCK scores are between 30 and 50. Among these 1188 pairs, 83 pairs of them, occupying about 6-7% of these pairs, form complexes with highest ZDOCK score ≥ 60 (Figure 5.14.). Their corresponding z scores were larger than 1.00. The ZDOCK score dependents to some extent on how strongly these pairs interact with each other. Therefore the pairs with outstanding ZDOCK scores were chosen as likely interacting CDK-cyclin complexes. These pairs are distributed inhomogeneously and centralized on several CDKs, for example AT3G48750, AT1G76540, AT2G38620 and AT1G67580. Most of them are phylogenetically closest to human CDK2 and CDK5. Judging from these ZDOCK scores they interact with multiple cyclins. This might be true in *Arabidopsis* but it is also very likely that these pairs are just false positives. Other

CDKs/cyclins display no obvious preference for interacting with any specific cyclins/CDKs. This might mean that their binding partners are not included in our proteins. However it is also very likely that because the appropriate template structures are not available for these proteins and their model structures are not reliable. Generally, our predictions are reliable for a subset of models for which suitable template structures are available.

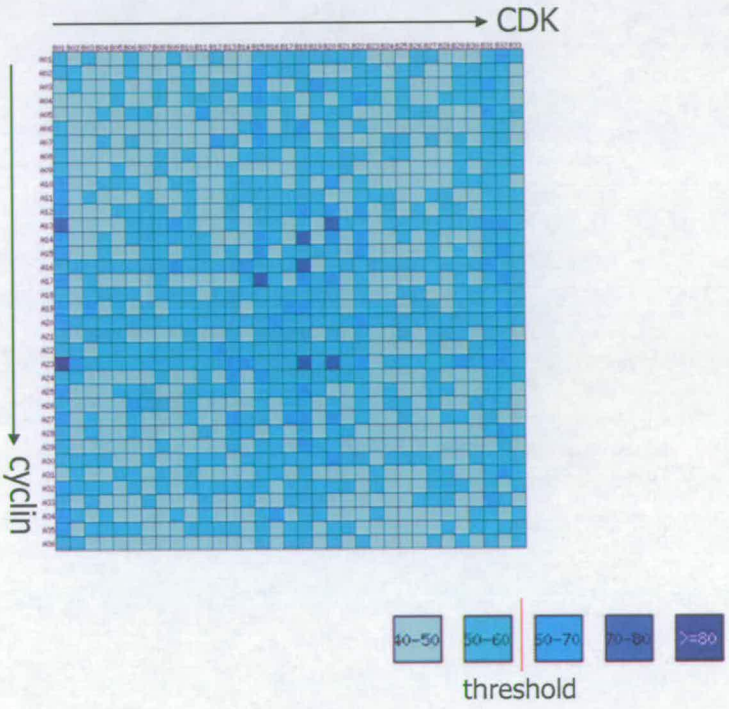


Figure 5.14. *Arabidopsis* CDK-cyclin Results: ZDOCK score Panel. The highest ZDOCK scores for 1188 pairs were stored in this panel. Each column represents a specific CDK-like protein and each row represents a specific cyclin-like protein. The colour in each cell displays the highest ZDOCK score of that pair. The darker the colour is, the higher the score. This picture was made by perl script. To see the codes for the CDKs and cyclins in this panel please refer to Appendix A.

ZDOCK score alone is not sufficient in practice for discriminating between true and false protein complexes. Automated protein-protein docking results typically contain too many false positive complexes to be directly used in practice. We therefore applied two additional criteria to select the most likely complexes from the ZDOCK result lists.

5.2.3.2 Subunit Orientation

The docking score is only a rough estimate of the association strength between two proteins in a certain association orientation. A high dock score does not necessarily refer to the correct subunit orientations, even when non-interface residues are already blocked as we still give ZDOCK a certain searching freedom. The Cα atom RMSDs between the CDK-cyclin complex structures generated by ZDOCK and reference structures (the crystal human CDK-cyclin complex structures) were calculated with program QUANTA. These RMSD values were then used as a criterion to evaluate relative subunit orientations in these docked complexes. Every pair's docked complex structure was superposed onto the three known human CDK-cyclin complex structures. The Cα atom RMSDs between docked complex structures and one of the three known CDK-cyclin complex structures (PDB entry 1FIN and 1H4L and model complex CDK6-cyclin D1) should be lower than or equal to 10 Å. The threshold was set as 10Å because the Cα atom RMSD between the three known human complex structures are between 10Å and 15Å. These pairs were also visually checked to make sure they are combined in the correct relative orientation.

Of the 83 pairs with ZDOCK score larger than 60, 59 have Cα atom RMSDs less than 10Å between their ZDOCK combined structure and the three human CDK-cyclin structures. All the pairs with outstanding ZDOCK scores which are larger than 70, were combined together in the correct subunit orientation. This fact can be treated as a support for our ZDOCK experiments.

Two pairs B01-A23 (Pair I) and B18-A16 (Pair II) were chosen as examples to give a detailed explanation of the subunit orientation selection criterion. The ZDOCK scores of both pairs are higher than 60; 85.95 for Pair I and 76.60 for Pair II. Their corresponding Z scores are also higher than 1.00. Figure 5.12 shows the superposed structures of the docked complex structures of Pair Ⅰ and Pair Ⅱ, and all the structures used in the positive control experiment (mentioned in Chapter 5.3.2). The Cα atom r.m.s.d.s between docked structures of Pair Ⅰ and Pair Ⅱ and crystal structure of human CDK2-cyclin A2 are 2.6 Å and 3.3 Å respectively. The slightly higher Cα atom RMSDs between these two pairs and crystal human CDK2-cyclinA2 mainly comes from the conformation divergence in the loop region of these structures. Pair I and Pair II are therefore both supported by correct relative subunit orientation.

5.2.3.3 Interface Property Criterion

The program MolSurfer (Gabadoulline *et al.*, 2003) is a graphical program that links the three-dimensional structure of a protein-protein complex with a two-dimensional projection map. Interface properties, including electrostatic and hydrophobic interactions, are projected to the two symmetrical 2-D interface maps and correlation coefficients of

surface properties between these two interface maps are then calculated. As electrostatic interactions and hydrophobic interactions are considered to be major factors to stabilize complexes, MolSurfer interface electrostatic correlation coefficient (ECC) and residue-residue hydrophobic correlation coefficient (HCC) can be used as interface property criteria to select potential CDK-cyclin pairs. We downloaded the standalone MolSurfer version and modified it (1) to calculate the number of pixels it divided each interface map into to provide an approximate estimate of the interface size, and (2) write the number of interface map size and all the correlation coefficients to a file so that large scale MolSurfer comparison can be run automatically.

Charges for atoms in PDB files were assigned by the AMBER force field of stand alone PDB2PQR (Dolinsky TJ. *et al*, 2004) software to work as input for the Poisson-Boltzmann Equation (PBE) solver in APBS ((Baker NA, *et a*l, 2001). PDB2PQR also add hydrogen atoms to the proteins. When it adds hydrogen atoms, it tries to ensure that the new hydrogen atoms are not rebuilt too close to existing atoms and also tries to optimize the hydrogen network. Grid files were then generated by APBS PBE solver. Finally, the PDB files and grd files for each chain in a complex, plus the PDB file of the complex, were input into the modified stand alone MolSurfer to calculate electrostatic and hydrophobic correlation coefficients. All these processes were automated by perl scripts.

1). Reference Sets: Transient Hetero-dimer Complex Set and Non-complex Set

CDK-cyclins are transient hetero-dimer complexes: that is complexes formed by different proteins which only interact transiently to carry out a particular biological task.

For calibrating the interface criterion we need two reference sets, a positive set which represent real transient hetero-dimer complexes, and a negative set of control data to represent false complexes, so that we could analyze interface property difference between true transient complexes and false complexes.

The positive set consisted of transient hetero-dimer complexes. The original transient heterodimer complex PDB entry name and the contacting residue numbers on interfaces between chains in each pdb file come from Yanay Ofran (Ofran & Rost, 2003). The sequences of these proteins were extracted from PDB using SRS7 system server (Zdobnov EM, *et al.*, 2002). Some protein sequences could not be extracted in this way and were abandoned. A set of criteria were applied to select chain pairs used in our further work: chain sequences need to be larger than 60 amino acids to avoid protein-ligand complexes; contacting residues between two chains in the complex should be more than 40 (two chains together) to get true complexes. Complexes with missing residues in their pdb files were discarded because missing residues may cause errors when using PDB2PQR to assign charges. Homologous sequences will cause bias to certain specific transient protein-protein interactions in the final results and so we need to filter out redundancy. PAM distances between these chain sequences were calculated using the DARWIN server. Most sequences are completely non-homologous (no sequence similarity was found). For sequences with PAM distance shorter than 130 (approximately equal to sequence identity percentage higher than 10 %), only one sequence was retained. The final criterion is the interface inaccessible surface area calculated by the Protein-Protein Interaction Server (http://www.biochem.ucl.ac.uk/bsm/PP/server/index.html). It has been observed that

natural protein-protein complexes usually have large interface. The interface inaccessible area of one partner in a complex should be larger than 600 $Å^2$ (Jones S. & Thornton J.M., 1995). So we set the interface inaccessible area size $\geqslant$ 600 $Å^2$ as another criterion in our selection. Finally we got 104 chain pairs that constitute the transient complex dataset.

The negative set, "non-complexes", was generated by using ZDOCK to combine proteins that normally do not interact. Here we chose 9 transient hetero-dimer complexes and 10 obligate complexes from the Yanay's dataset (Ofran & Rost, 2003). We searched these proteins online and to check whether there is any available information showing that these proteins might interact with each other. No such evidence was found. Based on this fact we ran all transient-complex-subunit by all obligate-complex-subunit docking experiments to generate false complexes. A set of very strict criteria were also applied to select non-complexes from the 360 output pairs to be fair for the transient complexes. Only the combined structures with ZDOCK score $\geq$ 60 and interface inaccessible surface area $\geq$ 600 $Å^2$ were selected as non-complexes. Finally we obtained 70 non-complexes to work as a negative reference dataset.

2). Two-Group Discriminant Function Analysis

ECC and HCC for each sample were calculated by modified standalone MolSurfer. These two variables were used to discriminate the transient set from the non-complex set.

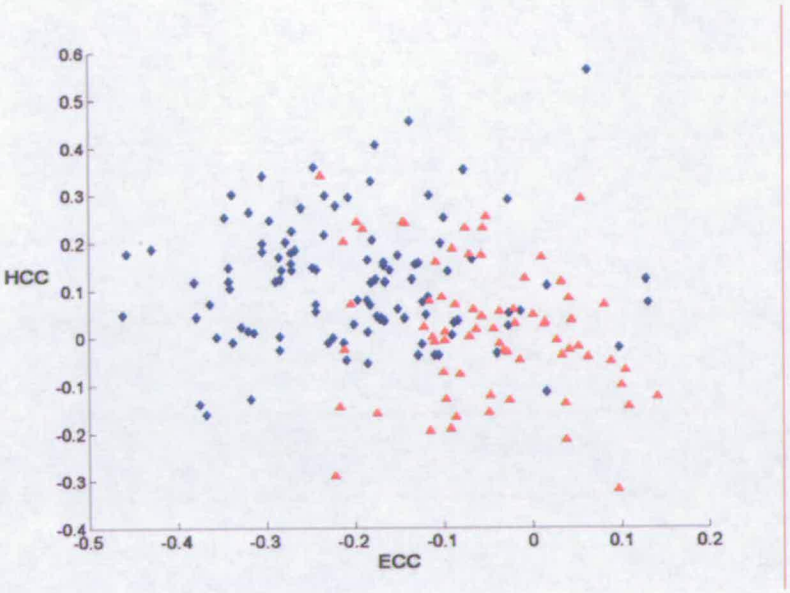


Figure 5.15 MolSurfer coefficients scatter plot for transient hetero-dimer complexes (blue) and non-complexes (red). These two groups are partly overlapped. ECC values discriminate these two groups better than HCC values.

From Figure 5.15 we can see that the means of ECC of these two sets are significantly different though there is still a certain overlap between these two reference sets (blue dots represent the transient complexes and red rectangular points represent non-complexes in this plot). Therefore we can use ECC variant to discriminate between these two groups. The mean of all the HCC values in these two reference sets are not significantly different. However, HCC does help to separate the two sets on the distribution plot. This is confirmed in the result of discriminant function analysis of these two variants. Our goal was try to find an optimal way to combine these two criteria to separate the two reference sets as best as we can.

Discriminate function analysis is a type of multivariate statistical analysis which generates new canonical functions that are linear combination of the mean-centred original variants. Here we run discriminate function analysis of all the ECC and HCC values of the two reference sets with program MATLAB (The MathWorks Inc). We got two canonical functions, CEH1 and CEH2 (combination of ECC and HCC). They can be illustrated by following equations:

$$CEH1 = (ecc - \overline{ecc}) \times e1 + (hcc - \overline{hcc}) \times f1;$$

$$CEH2 = (ecc - \overline{ecc}) \times e2 + (hcc - \overline{hcc}) \times f2;$$

Here 'ecc' and 'hcc' is the ECC and HCC for each complex, '$\overline{ecc}$' is the mean of eccs in these two groups, and '$\overline{hcc}$' is the mean of hccs in these two groups. "e1, e2, f1 and f2" come from eigenvec field that is a 2×2 real, non-symmetric array. This array defines the coefficients of the linear combinations of the original variables using inverse subspace iteration.

$$\text{Eigenvec} = \begin{bmatrix} e1 & e2 \\ f1 & f2 \end{bmatrix}$$

The function CEH1 provides the best discrimination between groups, and the second, CEH2, provides second best (Figure 5.16). In our study, e1 = 7.9812, e2 = 4.4613, f1 = -3.1498, f2 = 6.7870. The mean of ecc values and hcc values were -0.1402 and 0.0779, respectively.

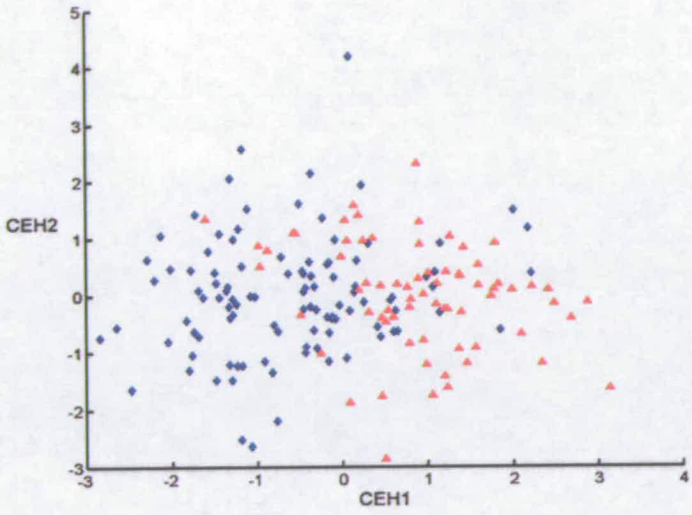


Figure 5.16 CEH1/CEH2 scatter plot for transient complexes (blue) and non-complexes (red).

The CEH1 and CEH2 axis were also projected onto the ECC-HCC plot (Figure 5.17) to get a visual impression about how CEH1 and CEH2 are related to the original variables ECC and HCC. CEH1 and CEH2 are not only enlargement of ECC and HCC in scale. The line on which CEH2s are equal to zero can be treated as the CEH1 axis in CEH1-CEH2 plot. And the line CEH1 equal to zero can be treated as the CEH2 axis. These two lines are not vertical and both leans a bit to the right direction. This reflects the fact that ECC is more weighted and therefore more magnified than HCC in the CEHs calculation.

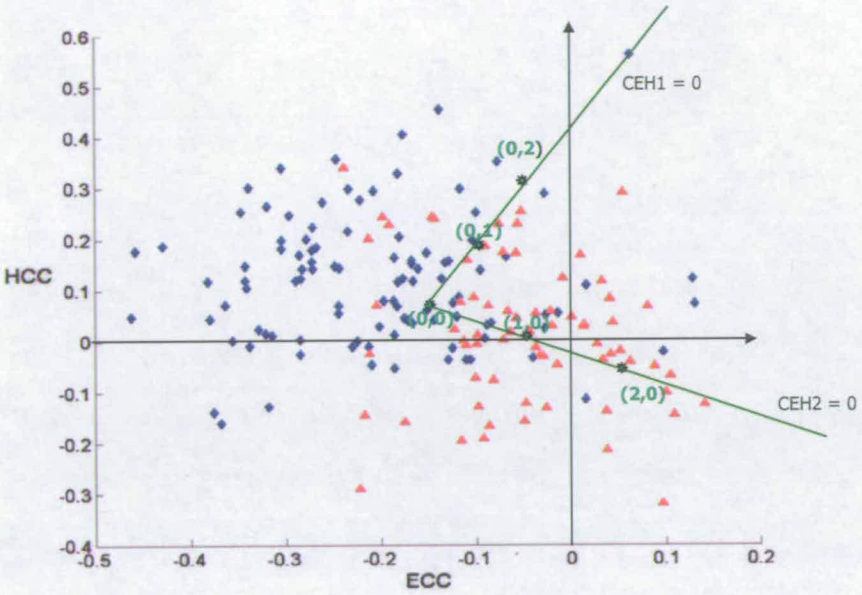
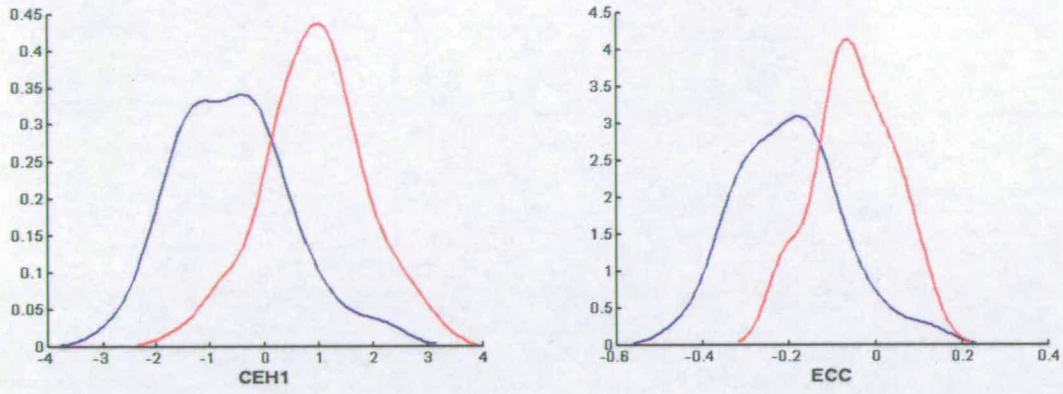


Figure 5.17 Project of CEH1 and CEH2 on the ECC-HCC plot. The two black lines represent the X- (ECC) and Y- (HCC) axis of ECC and HCC plot. The green line labeled with "CEH1 = 0" can be treated as the CEH1 axis in CEH1-CEH2 plot, and the green line labeled with "CEH1 = 0" can be treated as the CEH2 axis.

Gaussian distribution (normal distribution) of both CEH1 and ECC of these two reference sets generally all resemble the bell curve of standard normal distribution (Figure 5.18). CEH1 distribution curves are smoother and ECC distribution curves are more distorted. The area under Gaussian distribution curves stands for the probability density function (pdf). We can see that the percentage of overlapping area occupying the entire transient complex distribution area of CEH1 is lower (through the difference is not large) than that of ECC. This means that CEH1 do separate these two groups better than ECC alone and confirms the supposition that HCC helps a little in separating transient complexes from non-complexes.



A B

Figure 5.18. Gaussian Distribution of CEH1 (A) and ECC (B) for transient hetero-dimer complexes set (blue) and non-complex set (red).

3). Choice of cutoff position

In unusual cases with absence of overlap between distributions of different groups, there is no need for setting cutoff values but explicit value ranges to discriminate between different groups. In our cases the distributions of both CEH1 and ECC overlap between transient complex and non-complex groups. The choice of cutoff values involves consideration of both sensitivity and specificity. As displayed in Figure 5.19, from right to left in the region between arrow 2 and 1, the more to the left the cutoff position is chosen, the more specific and less sensitive the method will be to predict transient complexes. Similarly, from left to right in the region between arrow 2 and 3, the more to the right the cutoff position is chosen, the more sensitive and less specific the method is to predict transient complexes. Compared to position 1 which implies high specificity and position 3 which is very sensitive but with very low specificity, the cutoff position 2

is a balance between sensitivity and specificity and is chosen for our criterion calibration. Based on this strategy, the cutoff value of CEH1 is set as 0.138 and ECC is –0.13. With CEH1 being 0.138, 76.8% transient complexes are separated from non-complexes. With ECC being –0.13, 73.5% transient complexes are separated from non-complexes.

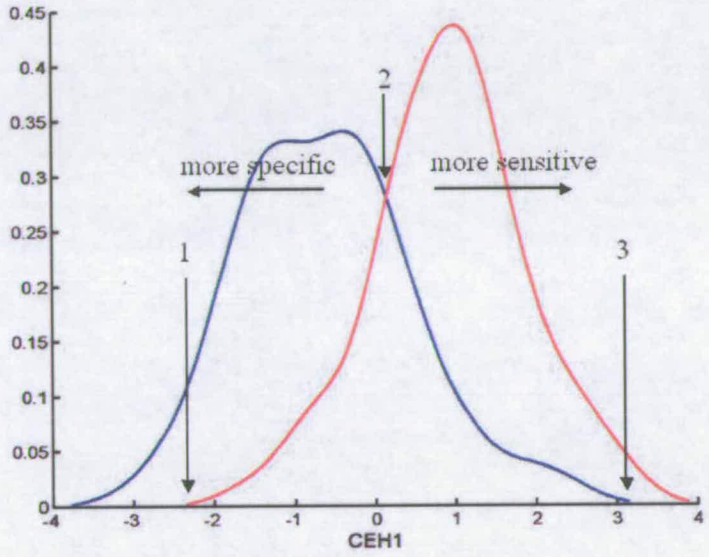


Figure 5.19. Choice of cutoff value position. The blue curve represents the Gaussian distribution of true transient hetero-dimer complexes; the red curve represents non-complexes. The crossover point of these two distribution curves is a statistically optimal cutoff position to separate the true transient complex group and non-complex group.

4). Cross-validation of CEH1 Criterion

To estimate the prediction accuracy for transient complexes and non-complexes using CEH1, we need both a training dataset and test dataset. A cross-validation was run by randomly selecting 80% of data from each reference set to work as training dataset and the remaining 20% as test data. Eigenvector values and ECC/HCC means of CEHs and

CEH1 cut-off value were derived from the 80% training data. Then these values were applied to calculate CEH1s for the other 20% testing complexes and also predict these complexes. The same calculation was repeated ten times. The prediction accuracies of around 80% (mean value: 80.6%; stdv: 0.0492)were obtained consistently with cutoff value of CEH1 around 0.13 (stdv: 0.0727). This supports the feasibility of using CEH1 to separate transient complexes from non-complexes.

5) Probability Ladder

Our predictions are mainly based on probability analysis. That is, the probability of the predicted CDK-cyclin pairs lying on the left of line c to be true is larger than 77% (Figure 5.20). If we set the cutoff value as -1.0 (line a), the probability of a potential complex located on line 'a' to be a true transient complex is eight times higher than the probability to be a false complex (non-complex). On the intermediate line 'b' where CEH1 value is -0.5 the probability of a pair to be true transient complex is 85%. Lines 'a', 'b' and 'c' divide the plot area into four subfields A, B, C, and D. The probabilities that a potential complex falling into area A is a true transient complex is above 90%; one falling into area B is between 85% and 90%, the one falling into area C is between 77% and 85%. If a potential CDK-cyclin complex falls into the area D, the probability that it is true transient complex is less than 77% and was not chosen as a likely CDK-cyclin pair. In this way, we divide the CEH1 distribution plot into a probability ladder from 90% to 85% to 77% above the cutoff value and the level below cutoff value being less that 77%.

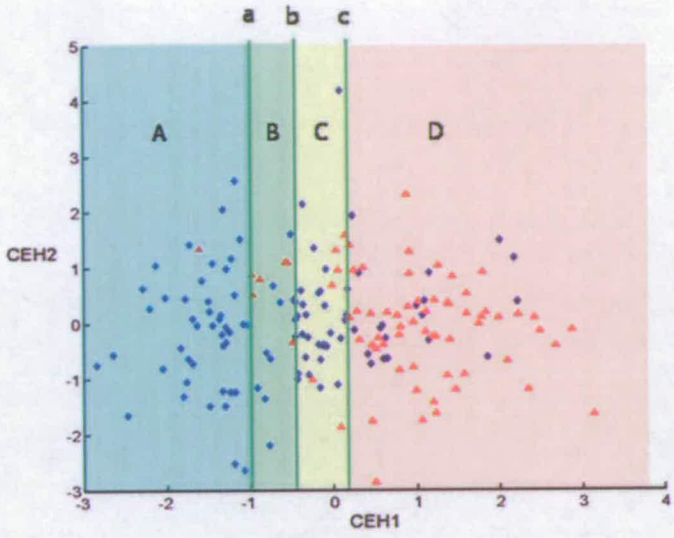


Figure 5.20 Distribution plot of the probability of a potential complex to be true transient complex. The CEH1 values are three vertical lines 'a', 'b' and 'c' are -1.0, -0.5 and 0.13. The probability of a potential complex located in area A to be true transient complex is above 90%. Similarly, one located in area B is between 85% and 90%, and one in area C is between 77% and 85%. To the right of line 'c', area D, the probability of a potential complex falling into this area being a true complex is less than 77%.

5.2.4 Application of Prediction Approach

1) Human CDK-cyclin Interactions

To test our prediction approach and estimate our prediction accuracy, we also modelled the structures of human CDK1-6 and cyclin A, B, D, E, H and P25 and then ran docking of all CDKs with all cyclins. These human CDK and cyclin sequences were extracted from Swiss-Prot (Bairoch A, *et al*, 2004). The multiple alignments of CDKs were built with T-Coffee. The N-terminal and C-terminal alignments of cyclins were extracted from Pfam (Bateman *et al*, 2004) and then manually joined together to work as target-

template alignments for model building. The same selection criteria were applied to the selection of human CDK-cyclin pairs. Experimental information about human CDK-cyclin interactions was extracted from Swiss-Prot and HPRD (Human Protein Reference Database) (Peri S., et al, 2003) and online google searching. No experimental information was available about which CDK will not interact with which cyclin. Therefore the cyclin-CDK pairs predicted to interact with each other in our approach were termed as "unconfirmed positive" interactions. Compared with the experimental data about human CDK-cyclin interaction (Table 5.3), the positive prediction (accuracy) of this prediction is around 82% (green cells ÷ (green cells + pink cells)) and coverage of this prediction is around 70% (yellow cells ÷ (green cells + yellow cells)). Based on the strict prediction and selection strategy, we assume that our *Arabidopsis* CDK-cyclin interaction prediction is about 80% accurate. The coverage of this prediction may be lower than 70% as we actually work with a subset.

	CDK1	CDK2	CDK3	CDK4	CDK5	CDK6
CGA1	+	+				
CGA2	+	+	+*			
CGB1	+	+	+			
CGB2	+	+				
CGD1				+	-	+
CGD2				+		+
CGD3				-		+
CGE1	-	-	-			
CGE2		-	+			
CGH						
P25					+	

Table 5.3. 3-D Dock Control Experiment: Human CDK-cyclin Interactions. Green: experimentally confirmed positive; Pink: unconfirmed positive; Yellow: false negative (information extracted from Swiss-Prot and HPRB). *: Meikrantz W. & Schlegel R. 1996.

2) *Arabidopsis* CDK-cyclin Interactions

As the main application of our prediction approach for transient hetero-dimer complexes, we predicted the interactions between *Arabidopsis* CDKs and cyclins: large scale docking was run between 33 CDK models and 35 cyclin models of *Arabidopsis*; Cα atom RMSD between ZDOCK combined complex structures and reference structures (known CDK-cyclin structures) should be less than 10Å, the CEH1 of possible *Arabidopsis* CDK-cyclin pairs ≤ 0.13. Finally we got 1188 potential CDK-cyclin pairs. 83 of them have outstanding ZDOCK scores (≥ 60). Among the 83 potential pairs, 59 of them were combined together with correct relative subunit orientation. All the potential pairs with outstanding ZDOCK score (larger than 70) were combined together in correct subunit relative orientation. This can be treated as a support of our use of ZDOCK to an extent. These 59 pairs are projected onto the CEH1-CEH2 plot of the two reference sets to see which are supported and which are not (Figure 5.21).

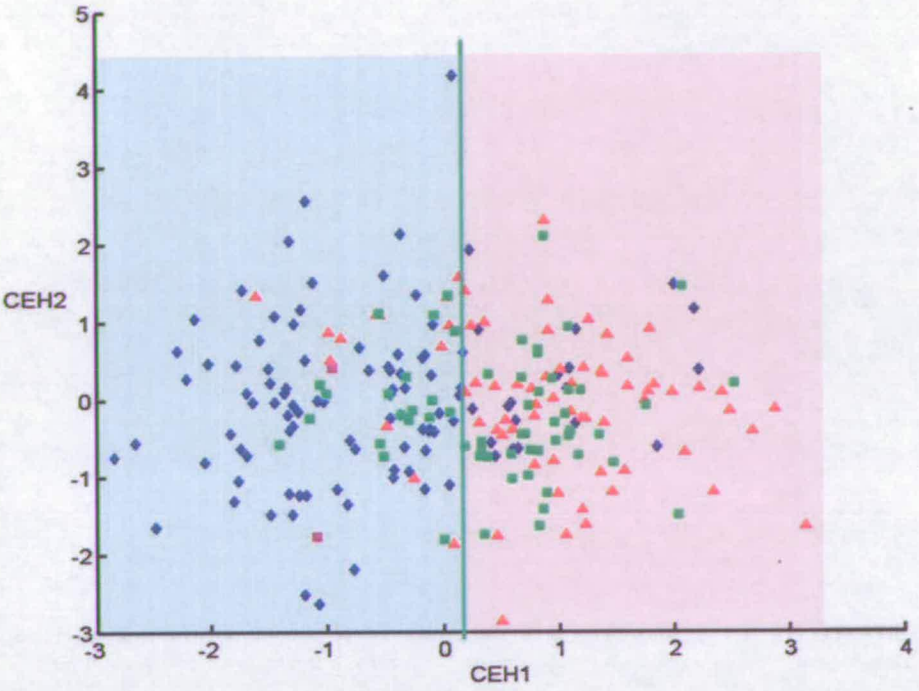


Figure 5.21 Scatter plot of *Arabidopsis* CDK-cyclin pairs. Transient reference complexes (positive set) are highlighted in blue, non-complexes (negative set) in red, two human CDK-cyclin complexes in magenta, and *Arabidopsis* complexes in green.

As can be seen in Figure 5.21, 19 of the 59 potential pairs are very likely to be true complexes (probability larger than 77%) and 40 pairs' probabilities to be true complexes are less than 77%. These 19 most likely CDK-cyclin pairs are listed in Table 5.4. In ascending order of CEH1, the most outstanding pair is formed between CDKA;1-CYCB1;4. Pairs CKL11-CYCA3;1 and CKL10-CYCB1;2 are in the second and third position of the list. The pair CDKA;1-CYCD3;1, which has been experimentally confirmed, is also ranked quite highly. If ranked according to ZDOCK score, pair

CDKA;1-CYCB1;2 would be at the top of the list. Most of these likely pairs are formed by A or B type CDK with A or B type cyclins. To an extent, their top-ranking may be related to our prediction only considering a subset as only a few template structures are available. All the CDK proteins' sequence identity percentages in these likely pairs to template sequences are above 40%. All these cyclin proteins' sequence identity percentages to template sequences are also above 30% except the CYCP family cyclin homologues which have only one five-helix repeat and low sequence identity to template sequences. The quality evaluation results given by the four programs (What_check, Procheck, ERRAT, and Prove) are satisfactory for all the CDK models, and satisfactory or acceptable for all these cyclin models with one exception, AT5G07450. The quality evaluation of model AT5G07450 with program Pro_check, ERRAT and What_check all got satisfactory results except in Verify_3D evaluation.

CDK locus name	Gene	Cyclin locus name	Gene	CEH1	ZDOCK score	ECC
AT3G48750	CDKA;1	AT2G26760	CYCB1;4	-1.4134*	67.79	-0.30706#
AT1G09600	CKL11	AT5G43080	CYCA3;1	-1.1489*	61.44	-0.26538#
AT1G57700	CKL10	AT5G06150	CYCB1;2	-1.0552*	69.87	-0.23569#
AT1G67580	CDKG;2	AT5G07450	CYCP4;3	-1.0074*	61.76	-0.23645#
AT1G67580	CDKG;2	AT2G45080	CYCP3;1	-0.5503*	67.09	-0.14348#
AT5G63610	CDKE;1	AT2G26760	CYCB1;4	-0.5447*	61.20	-0.22034#
AT3G48750	**CDKA;1**	**AT4G34160**	**CYCD3;1**	**-0.5118***	**64.25**	**-0.22443#**
AT3G48750	CDKA;1	AT5G06150	CYCB1;2	-0.4679*	85.95	-0.18311#
AT2G38620	CDKB1;2	AT1G47220	CYCA3;3	-0.3635*	70.26	-0.18494#
AT3G48750	CDKA;1	AT3G11520	CYCB1;3	-0.3209*	63.74	-0.15814#
AT5G63610	CDKE;1	AT4G37490	CYCB1;1	-0.2983*	67.74	-0.18175#
AT1G67580	CDKG;2	AT3G21870	CYCP2;1	-0.267*	63.25	-0.17282#
AT3G54180	CDKB1;1	AT1G76310	CYCB2;4	-0.1149*	60.32	-0.16176#
AT5G63610	CDKE;1	AT4G03270	CYCD6;1	-0.0946*	60.69	-0.14989#
AT5G63610	CDKE;1	AT1G47220	CYCA3;3	-0.0777*	68.34	-0.09719
AT1G67580	CDKG;2	AT1G80370	CYCA2;4	0.0036*	70.27	-0.22301#
AT1G76540	CDKB2;1	AT5G07450	CYCP4;2	0.0397*	66.48	-0.07409
AT3G48750	CDKA;1	AT1G76310	CYCB2;4	0.0531*	65.08	-0.14224#
AT3G54180	CDKB1;1	AT3G11520	CYCB1;3	0.1048*	69.70	-0.08891
AT2G38620	CDKB1;2	AT1G47210	CYCA3;2	0.187	61.65	-0.1491#
AT1G76540	CDKB2;1	AT1G47230	CYCA3;4	0.309	71.45	-0.1428#
AT3G48750	CDKA;1	AT1G20610	CYCB2;3	0.3285	67.10	-0.13228#
AT5G63610	CDKE;1	AT1G80370	CYCA2;4	0.33	62.02	-0.13753#
AT1G20930	CDKB2;2	AT1G77390	CYCA1;2	0.3467	67.51	-0.18595#
AT1G76540	CDKB2;1	AT1G80370	CYCA2;4	0.3774	62.02	-0.13667#
AT2G38620	CDKB1;2	AT3G50070	CYCD3;3	0.8211	66.81	-0.13385#

Table 5.4 Most likely interacting *Arabidopsis* CDK-cyclin pairs selected by ZDOCK, their associated subunit orientation and CEH1 values

*: Pairs supported by CEH1 value. #: Pairs supported by ECC value.

The pair highlighted in green has been confirmed experimentally (Healy JMS, *et al*, 2001).

3) CDK-cyclin Interactions in *Trypanosomatid Brucei* and *Leishmania Major*

T.brucei and *L.major* are two pathogens that cause severe diseases to human. A number of CDK and cyclin homologues have been discovered in these two organisms: eleven cyclin-like sequences in *L.major*, ten cyclin-like sequences in *T.brucei*, and twelve CDK-like sequences in both *L.major* and *T.bruce* (Naula C. *et al*, 2005). Christina *et al* have tried to classify these sequences based on their sequence characteristics or their sequence similarity to animal CDKs and cyclins (Table 5.5).

Name	"PSTAIRE" motif	*L. major*	*T. brucei*
CRK1	PCTAIRE	LmjF21.1080	Tb10.70.7040
CRK2	SVSSIRE	LmjF05.0550	Tb07.30D13.430
CRK3	PQTALRE	LmjF36.0550	Tb10.70.2210
CRK4	PGAAIRE	LmjF16.0990	B08.5H5.130
CRK5	QVNRLRE	LmjF35.5010	Tb09.211.0960
CRK6	PATTIRE	LmjF27.0560	Tb11.47.0031
CRK7	PHPVARE	LmjF26.0040	Tb07.43M14.340
CRK8	HRCTFRE	LmjF11.0110	Tb11.02.5010
CRK9	QREEARP	LmjF27.1940	Tb927.2.4510
CRK10	RKGAFDA	LmjF29.2150	Tb03.48K5.160
CRK11	SATVLRE	LmjF30.1780	Tb06.5F5.880
CRK12	PQTSLRE	LmjF09.0310	Tb11.01.4130

A

(to be continued)

Name	Products	*L. major*	*T. brucei*
CYC4	Putative CYC2-like cyclin	LmjF05.0710	Tb07.21H15.170
CYC11	Putative CYC2-like cyclin	LmjF24.1880	Tb08.11J15.300
CYC10	Putative CYC2-like cyclin	LmjF24.1890	Tb08.11J15.340
CYCA	Putative mitotic cyclin	LmjF25.1470	-
CYC8	Putative mitotic cyclin	LmjF26.0330	Tb07.27M11.950
CYC3	Putative mitotic cyclin	LmjF30.0080	Tb06.3A7.1310
CYC7	Putative CYC2-like cyclin	LmjF30.3630	Tb06.30P15.430
CYC9	Putative cyclin C-like	LmjF32.0760	Tb11.01.5600
CYC2	Putative CYC2-like cyclin	LmjF32.0820	Tb11.01.5660
CYC6	Mitotic cyclin experimentally characterized	LmF32.3320	Tb11.01.8460
CYC5	Putative CYC2-like cyclin	LmjF33.0770	Tb10.26.0510

B

Table 5.5 CDC2-related (CDC2 is the old name of CDK1 and is more commonly used in the biologist community) kinases (A) and cyclins (B) predicted from the genome sequencing of *Leishmania Major* and *Trypanosoma Brucei* (Naula C, *et al*, 2005).

We applied our entire modeling and prediction approach to these CDK-like and cyclin-like sequences. These CDK and cyclin homologue sequences were extracted from the database GeneDB (Hertz-Fowler C *et al*, 2004). The multiple alignments of *L. major/ T. brucei* CDK homologue sequences and template (human) CDK sequences were built with the program T-Coffee. The range of sequence percentage identity between target and template sequences was between 40% and 50%. The multiple alignment of target cyclin homologue sequences from *L. major/T.brucei* and template (human) cyclin sequences were built with the HMM tool provided in database SUPERFAMILY. The low sequence identities between the cyclin like sequences of these two species to human

cyclins, ~10%, induced great difficulty in generating alignments with high quality and therefore greatly affect the quality of the 3-D comparative models of these sequences. This was reflected in poor all-by-all docking results, as the highest ZDOCK scores of these CDK-cyclin combinations were mostly very low with only very few over 60. Accordingly, our prediction coverage was expected to be very low in such difficult cases. However, the positive prediction accuracy should not be greatly changed because of our set of selection criterion calibrations and control experiments. In fact, of the combinations with ZDOCK score over 60, we got two pairs, CRK3-CYC6 and CRK3-CYC10, with correct relative subunit orientations and high interface electrostatic complementarity (Table 5.6). This pair has been confirmed by experiment (Naula C. *et al*, 2005). CRK3-CYC6 has also been co-crystallized (personal communication with Malcolm Walkinshaw).

CDK Name	CYC Name	ZDOCK score	ECC	HCC	CEH1
CRK3	CYC6	71.40	-0.2233	0.1486	-0.6278
CRK3	CYC10	61.42	-0.1321	0.0668	-0.1578

Table 5.6 Two likely interacting CDK-cyclin pairs predicted through our approach in *T.brucei*.

5.3. Most likely negative *Arabidopsis* CDK-cyclin pairs

Since interface property criteria make a reasonably clear discrimination between transient complexes and non-complexes in our reference sets, it may also be feasible for us to use them to predict likely negative CDK-cyclin pairs. This work is done by running large scale MolSurfer analysis of all the 1188 potential *Arabidopsis* CDK-cyclin pairs combined by ZDOCK. Then these pairs were ordered according to their ECC

values. Pairs with high ECCs and low HCCs were selected to be most likely negative complexes (Table 5.7). These likely negative pairs are mainly formed between CKL/MPK family members and CYCP/A/B family members. There is currently no information available to validate our negative predictions.

CDK locus number	Gene	Cyclin locus number	Gene	ZDOCK score	ECC	HCC
AT5G39420	CKL1	AT3G21870	CYCP2;1	41.25	0.1403	-0.195
AT1G66750	CDKD;2	AT2G44740	CYCP4;1	54.28	0.15	0.008
AT1G53050	CKL11	AT2G45080	CYCP3;1	48.58	0.147	0.015
AT4G36450	MPK14	AT5G25380	CYCA2;1	46.76	0.149	-0.085
AT1G10210	MPK1	AT4G37490	CYCB1;1	53.56	0.181	-0.106
AT1G18670	CKL3	AT4G35620	CYCB2;2	46.74	0.193	0.072
AT3G48750	CDKA;1	AT3G21870	CYCP2;1	49.71	0.127	-0.125

Table 5.7 Most likely negative *Arabidopsis* CDK-cyclin pairs.

5.4 Validation Studies

5.4.1 Interface Polar Percentage

It is quite straight forward to assume that there should be a linear relationship between interface polar percentage (the percentage of polar interface residues of the total interface residues) and electrostatic correlation coefficient. We randomly selected 37 transient complexes, 10 obligate complexes from Ofran & Rost's dataset and 11 non-complexes to analyze this potential relationship. Polar residue percentages in interfaces were calculated by perl scripts. Interface residues were defined as residues on one protein whose side-chain atoms (non-hydrogen) are in 5Å distance with the non-hydrogen side-chain atoms of any residues on another protein in a complex. Polar amino

acids include Cys, Asn, Gln, Ser, Thr, Arg, His, Lys, Asp, Glu. The values of these complexes were compared to their electrostatic correlation coefficient values (Figure 5.22). No obvious linear correlation can be observed between ECC and interface area size or polar percentage.

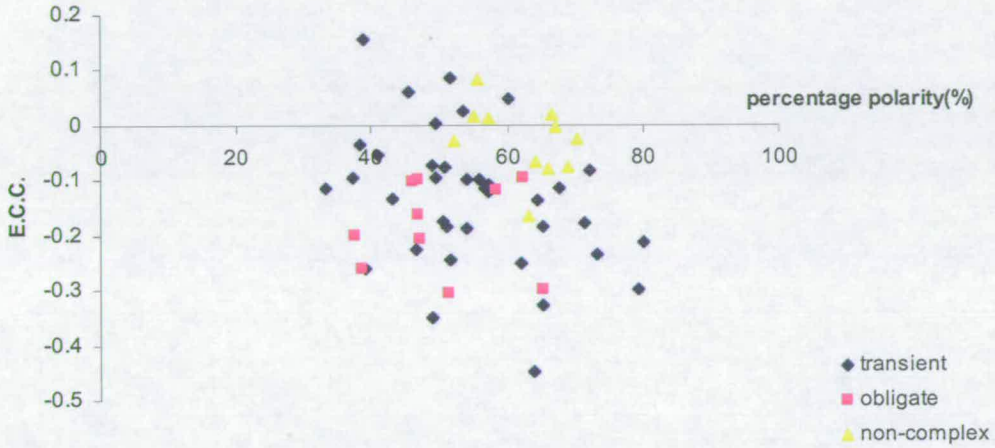


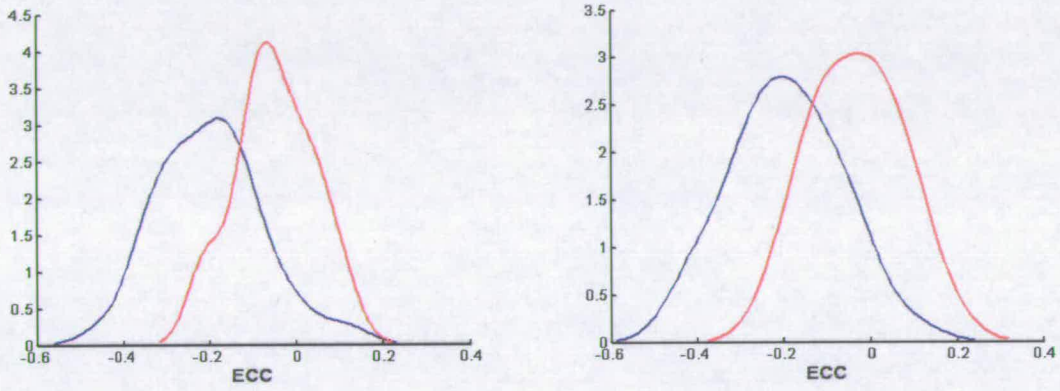
Figure 5.22 Interface percentage polarity and ECC

5.4.2 Force-Field Dependence

When calculating ECCs of complexes, charges of each atom in the PDB files need to be assigned and hydrogen atoms need to be added. This work was done by force field through the program PDB2PQR. Different force fields have different parameters and weightings for charges and hydrogen networks (hydrogen atom positions). We use two well-known force fields, AMBER and CHARMM, to analyze their different affect on the final result.

The extent to which transient complexes and non-complexes are separated is quite similar in both the analyses (Figure 5.23). With the CHARMM ECC cutoff value being -

0.14, 67% transient complexes fall into the positive (real complex) field and 79% non-complexes fall into the negative (false complex) field. With the AMBER ECC cutoff value being –0.13, 74% transient complexes fall into the positive field and 79% non-complexes fall into negative field. Generally, AMBER ECCs give better separation than CHARMM ECCs.



A B

Figure 5.23. The Gaussian distributions of ECCs for transient hetero-dimer complexes and non-complexes using force-field AMBER (A) and CHARMM (B) separately.

5.4.3 Reproducibility of force field and CEH1 Error Bar

Force fields usually try to optimize the protein hydrogen network and the water hydrogen network when assigning charges to each atom in a protein. The PDB2PQR program uses a Monte Carlo searching while optimizing the global hydrogen bond network in the protein. This step is rate-limiting and is not always possible for the program to find the best solution for hydrogen network. This might affect the final ECC value of large protein complexes.

We tested the Amber force-field reproducibility by repetitively (10 times) running PDB2PQR to generate PQR files with protein 1FIN, 1AVA, 6PRC and 1H4L. The reproducibility of the force-field was reflected by the ECC values calculated by MolSurfer. The standard deviation of these proteins' ECC values varies from 0.02 to 0.04 (Table 5.8).

Structure	1fin	1h4l	1ava	6prc
average ECC	-0.3122	-0.2144	-0.0852	-0.0445
STDV	0.0208	0.0250	0.0401	0.0296

Table 5.8 The mean values and standard deviations of ECCs of four protein complexes.

As HCC is highly reproducible, the standard deviation of CEHs could be directly deduced from ECC's standard deviation.

Stdv (CEH1) = e1 × stdv (ecc) – e1 × stdv ($\overline{ecc}$) = ±0.32

Because the distribution of ecc values is nearly random in our reference sets, the standard deviation of all the $\overline{ecc}$ s should be nearly zero and could be ignored. Therefore, the standard deviation of a CEH1 is equal to the standard deviation of an ecc multiplied by e1. In this way, we estimated the error bar of CEH1 around the cutoff value (Figure 5.24). The dots falling into the grey area of error bar could not be classified confidently onto a certain probability level. The most divergent ecc's standard deviation was used to calculate the cut-off CEH1 value's standard deviation.

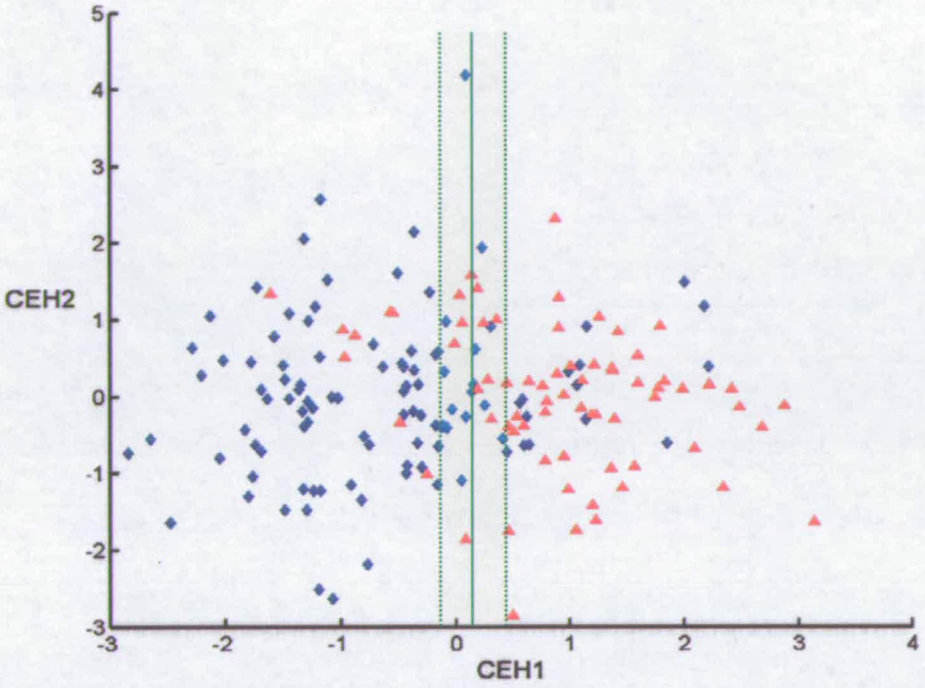


Figure 5.24 Error bar of CEH1 at the cutoff position

5.4.4 Model and Dock Complex Structures

Because we used model structures in the docking program, model quality may have had a great impact on the docking result. The differences between docked complexes, and model complexes were studied by comparing their interface surface area sizes and the Cα RMSDs between docked/modelled complex structure and crystal complex structure of the same two proteins. We randomly selected ten structures from our transient hetero-dimer complex reference set. Docked complexes are recombined complex structures (combined by ZDOCK) between two crystal chains. Modelled complexes are structures built by MODELLER onto the crystal complexes themselves. Generally, the docked complex interfaces were larger than in the crystal complexes, and the interfaces in the

crystal complexes larger than in the modelled complexes. So it seems that the docking program (ZDOCK) puts the two proteins in a complex closer together, and Modeller a little further apart. The Ca RMSDs between crystal complexes and docked complexes are slightly higher than those between crystal complexes and modelled complexes (table 5.9).

protein	Crystal Complex	Dock Complex		Model Complex	
PDB ID	Interface Size (Å^2)		r.m.s.d. (Å)		r.m.s.d.(Å)
1a0o	578/550	725/648	1.21	581/534	0.18
1am4	854/954	960/1072	0.65	881/946	1.12
1ava	1243/1264	1294/1310	0.57	1242/1264	0.15
1b6c	891/869	1028/970	0.55	836/831	0.25
1a2k	733/843	812/1001	0.77	723/815	0.28
1ahj	3765/3747	3846/3806	0.84	3690/3607	0.15
1aro	1087/1008	1313/1151	0.63	961/1092	1.19
1bou	2041/2150	2129/2197	0.53	2017/2145	0.18
1a4y	1280/1378	1339/1411	0.55	1294/1407	0.25
1eth	744/832	735/812	0.41	761/837	0.18
1iar	733/822	856/979	0.73	699/781	0.73
2mll	1163/1194	1449/1478	0.94	1127/1209	0.30
2wsy	929/838	1022/953	0.74	913/822	0.56
1fin	1610/1794	1716/1904	0.80	1625/1766	0.79
1h4l	1491/1488	1548/1559	0.54	1467/1445	0.63
6prc	4284/4270	4122/4141	0.87	4259/4258	1.78
7cei	677/707	837/840	1.58	726/775	0.20

Table 5.9 Model and Dock Complex Structures Compared to Crystal Complex Structures.

5.4.5 Feasibility of Large Scale Modelling Approach

Another possible prediction approach is to run large scale modelling to combine the structure of *Arabidopsis* CDKs and cyclins together in a pair-wise way and then use interface property criteria to select the most likely pairs. Theoretically this should be feasible. The problem is that the limited available template structures and the slight divergence of relative orientation between CDKs and cyclins make the modeling of CDK-cyclin complexes not very reliable. Another problem we met was that all the modeled complexes have low ECC values, even for the very negative pairs combined by ZDOCK. This means that the MolSurfer interface property criterion alone would probably not work well here to discriminate true complexes from false complexes.

6.

DISCUSSION

6.1 Prediction Approach

6.1.1 Are comparative models useful?

Although comparative models can be very accurate when the sequence similarities between target and template structures are high (percentage sequence identity larger than 40%), they still contain errors, especially the side-chain atom conformation. Some work has been done to see whether comparative models can be used for protein function prediction, for example, protein-protein docking. Chakravarty et al (Chakravarty S *et al*, 2005) compared the structure-derived properties of about twelve thousand single-template comparative models with their NMR or X-ray template structures. They found that the difference of most of the structure-derived properties between comparative models and NMR/X-ray structures is in the same order as the difference between NMR and X-ray structures. Comparative models' surface areas are generally bigger than experimental structures as they are more rugged. And including solvent effect during model building or refinement might help to improve the accuracy of surface properties of comparative models.

6.1.2 Could the ZDOCK scoring scheme be improved according to our analysis?

In our prediction procedure, ZDOCK discriminate 83 pairs from 1188 pairs and 59 of them were combined together with correct subunit orientation. However, through MolSurfer analysis, we still eliminate most of them and only get 19 most likely

interacting pairs. All this may suggest that MolSurfer separates transient complexes from non-complexes better than ZDOCK and the ZDOCK scoring function could still be improved, probably through new electrostatic interaction calculation. This may be true particularly for CDK-cyclins. There are not enough data to pass judgement.

6.1.3 Could this prediction approach be applied to other protein-protein interactions?

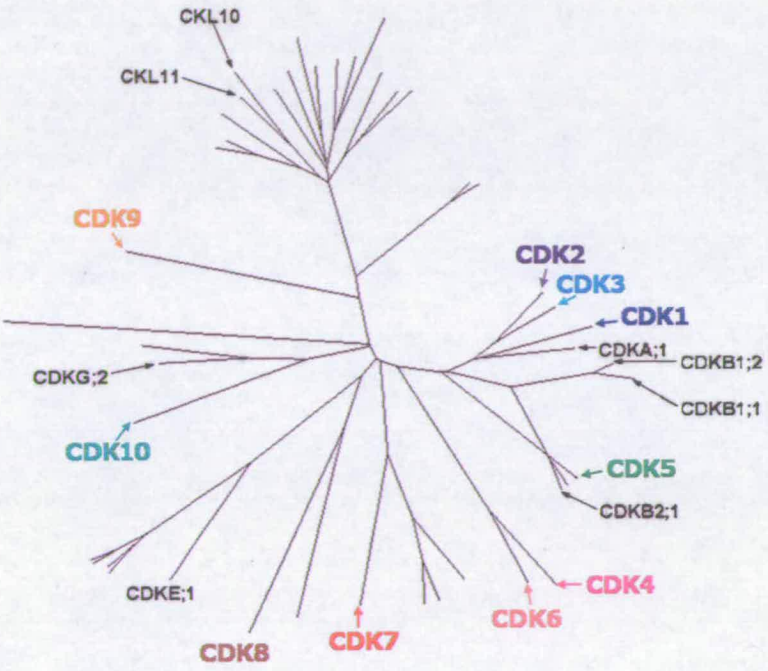
In our calibration of the interface property criteria, not only known CDK-cyclin complex structures but also other transient hetero-dimer complexes were investigated. These transient complexes were also non-homologous with percentage sequence identity lower than 10%. Therefore, the derived interface property criterion cutoff value should be applicable to all transient heterodimer complexes to separate them from non-complexes. Whether our prediction approach can be applied to other specific protein-protein interaction will mainly depend on the availability of their 3-D structures or their close homologues' known 3-D structures. Also this strategy is only applicable to cases in which the location of the binding sites (with respect to structure) is conserved.

6.1.4 Weakness and Utility of this Prediction Approach

The entire prediction approach is computationally automated, including the calibrations and calculations. The running time for large scale docking could be reduced by using more powerful computers. The bottleneck of this prediction approach is that making good models takes time and requires manual work and personal experience. This limits the transferability and automation of this strategy. This may be improved with the emergence of more accurate alignment generation programs and modelling programs.

6.2 Analysis of Predicted Pairs

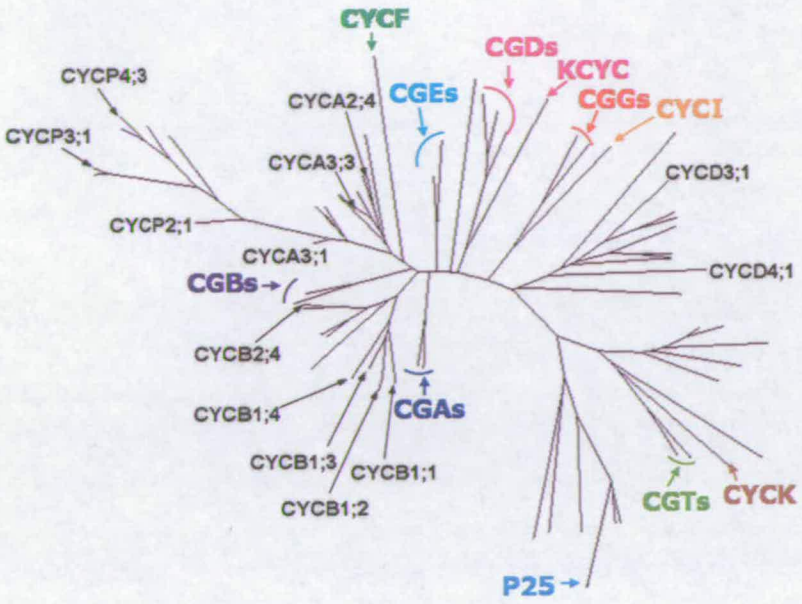
6.2.1 Are the 19 *Arabidopsis* most likely interacting CDK-cyclin pairs predicted in our approach also phylogenetically feasible?



Tree fitting Index = 0.67

A

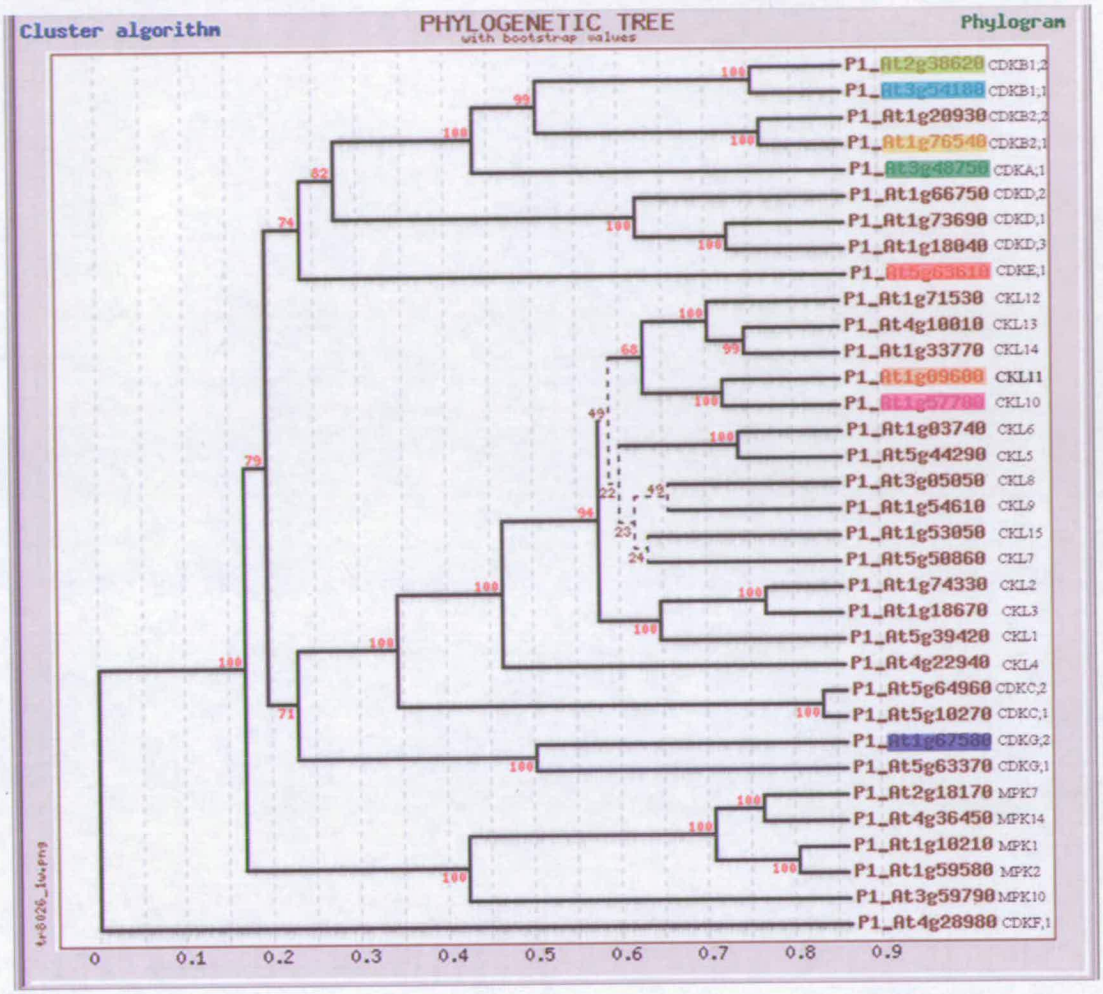
(to be continued)



Tree fitting Index = 0.76

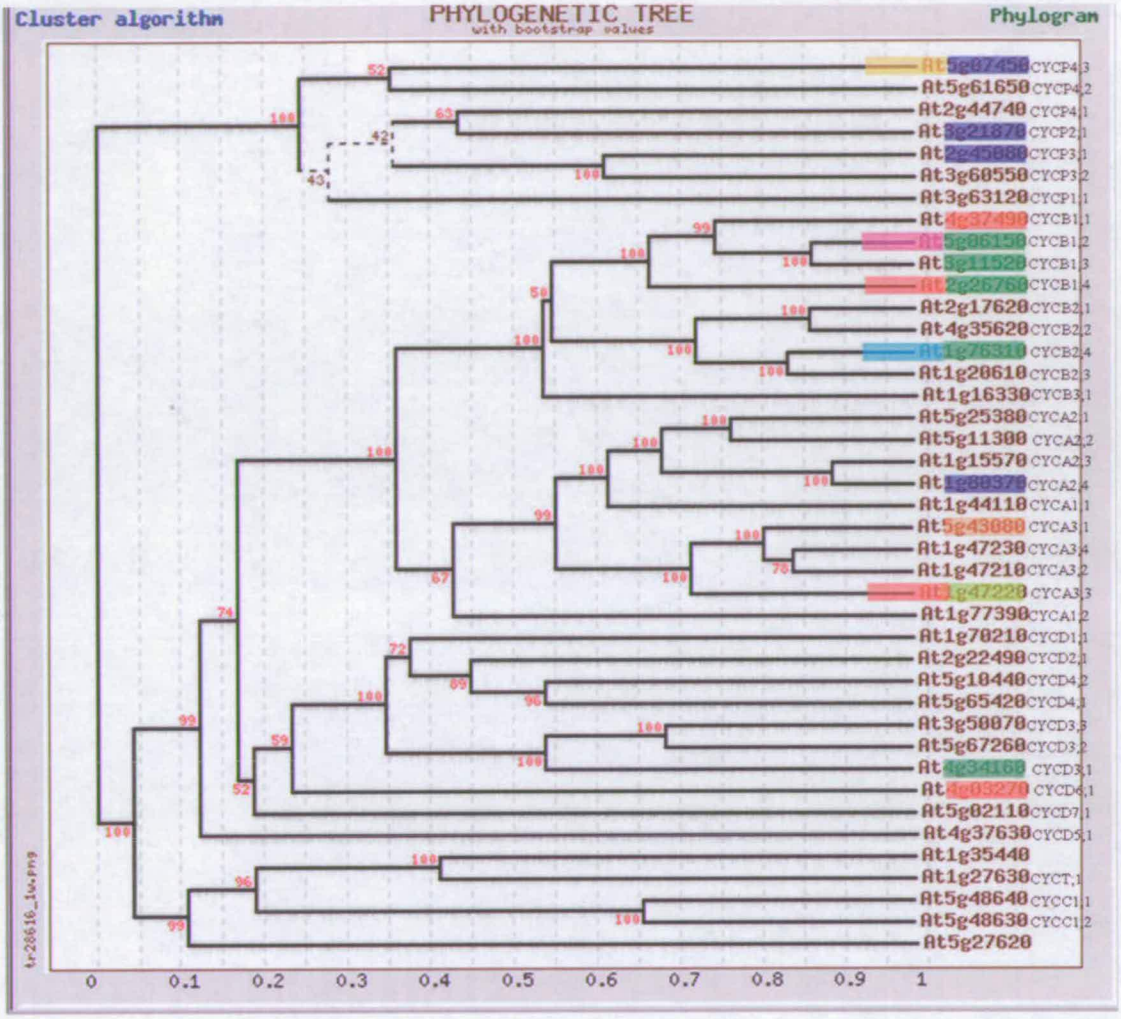
B

Figure6.1. A- unrooted phylogenetic tree built by DARWIN server of *A.thaliana* and human CDK homologues; B- unrooted phylogenetic tree built by DARWIN of *A.thaliana* and Human cyclin homologues. All the human CDKs/cyclins have been labeled using different colors (except black). The *Arabidopsis* CDK/cyclin like sequences appearing in the predicted 19 most likely pairs are labeled in black. For tree fitting index, poor is > 1, good is < 1.



A

(to be continued)



B

Figure 6.2. The Phylogenetic Trees reconstructed by TreeTop web server (http://www.genebee.msu.su/services/phtree_full.html) based on the multiple alignments finally used for comparative modelling. The so-called CLUSTER algorithm was used and bootstrapping requested. Tree A is the tree of *A.thaliana* CDK-like sequences. Tree B is the tree of *A.thaliana* cyclin like sequences having all the two cyclin-folds. In Tree A, the CDKs appearing in the final 19 mostly likely pairs were highlighted in different colors. The cyclins in each pair were then highlighted in their associated CDKs' colors.

The 20 most likely pairs are listed in Table 5.3. Most of these pairs are formed by A and B type CDKs with A/B/D type cyclins （Figure 6.1, Figure 6.2）. CDKA;1 interacts mainly with CGB type cyclins. CDKE;1 interacts with CGB, CGA and CGD type of cyclins. The CDKA;1-CYCD3;1 pair has been confirmed by experimental data (Healy J.M.S. *et al* 2001). Due to limited template structures available and relatively high sequence diversity between A, B type cyclins and other type cyclins in both animal and plant, our predictions considered a subset mainly restricted to the plant orthologues of human CDK1, 2, 3, 4 and cyclin A, B, D. No plant orthologues of human CDK5 and CDK6 have been found yet. This may explain to an extent why our predicted pairs are mainly CDKA/B-CYCA/B/Ds.

The CDK in Pair Ⅰ, CDKA;1, is most similar to mammalian CDK 1, 2- and 3- and has the same canonical PSTAIRE motif (Figure 6.1). This CDK has been shown to regulate both the G1-to-S and G2-to-M transitions (Segers G *et al*, 1996). The cyclin in Pair Ⅰ is CYCB1;2, which is most similar in sequence to mammalian cyclin B. The CYCB RNAs are known to be present in G2 and M phase (Shaul O, et al, 1996; Kouchi H, et al, 1995). Therefore, based on phylogeny location and expression time, Pair Ⅰ is a very likely complex. Similarly CDK in PairⅡ, *Arabidopsis* CDKB2;1 which can be identified by the presence of PSTTLRE/PPTTLRE (for CDKB2 type and for CDKB1 type will be PPTALRE) motif, is a plant-specific gene CDK class. B type CDKB2 is expressed only in G2-to-M cells (Menges M *et al*, 2002). CYCA1;2, the cyclin in Pair Ⅱ, its RNAs increase at or after the onset of S phase till G2 and occasionally M phases (Fuerst RAUA, *et al* 1996; Ito M. *et a*l, 1997).

From the ZDOCK score panel (Figure 5.14), it is clear that some CDKs, for example the CDK in Pair I, AT3G48750 (CDKA;1), prefer to interact with many cyclins. Other CDKs like B09-AT1G20930 (CDKB2;2), B15-AT1G67580 (CDKG;2), B18-AT1G76540 (CDKB2;1), B20-AT2G38620 (CDKB1;2), B32-AT5G63610 (CDKE;1) display similar properties. Some cyclins, for example A13-AT1G77390, A14-AT1G80370, A15-AT2G26760, A19-AT4G37490, A23-AT5G43080, A27-AT3G50070, also prefer to interact with more CDKs than other cylins. These behaviours might have special biological meaning.

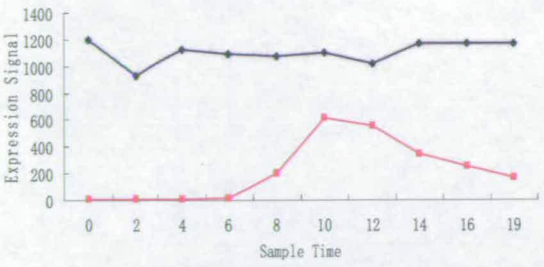
Not all the CDKs in *A.thaliana* have orthologues in human. The inverse is also true. For example, it is not obvious which is the closest orthologue of human CDK4, CDK6 and CDK9 according to the phylogenetic tree of human and *Arabidopsis* CDK homologues. Some *Arabidopsis* CDK sequences also do not seem to have clear orthologues in human. This is because some CDKs are animal specific, for example CDK6, and some CDKs, for example CDKB, are plant specific. Compared with human, *Arabidopsis* has many more CDK homologues and cyclin homologues. Unlike human, *Arabidopsis* possesses more than one representative of a given CDK or cyclin. Why *Arabidopsis*, and other plants, need so many CDKs and cyclins might be of special biological meaning and needs further investigation.

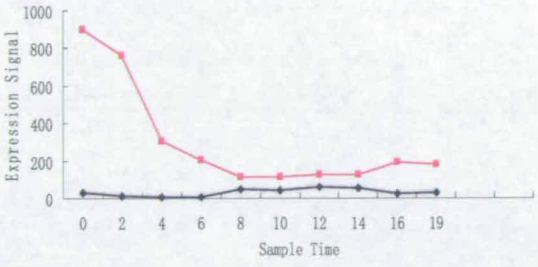
6.2.2 Expression Profile Analysis of the 19 Most Likely Pairs

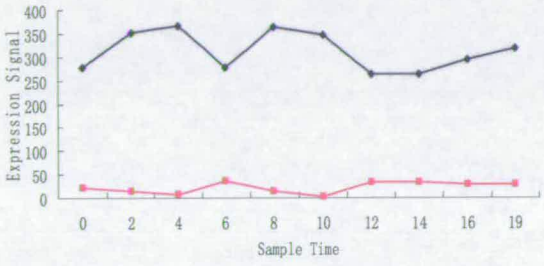
It is expected that the CDK protein levels in plant cell should also be stable while the levels of cyclins fluctuate at different cell cycle phases, just like the regulations of CDKs and cyclins in human cells. That is, at different phases of the cell cycle, different cyclin

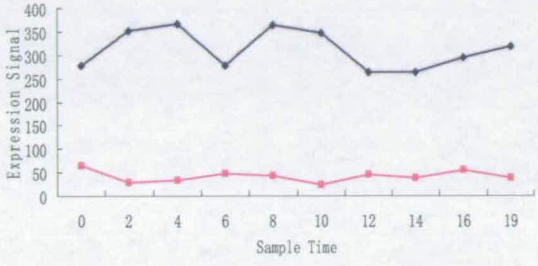
partners' concentration might greatly rise or fall so that different CDK-cyclin complexes could be formed or broken. However, one CDK can interact with several different cyclin partners (at the same stage of cell cycle and at different stage of cell cycle), and one cyclin can also interact with several different CDK partners. The expression profiles of CDK-cyclins can therefore be complicated and diverse. Generally the only statement we can make is that, if one CDK and one cyclin are not co-expressed at any time point of the cell cycle in any tissue, they will probably not be able to interact with each other in nature.

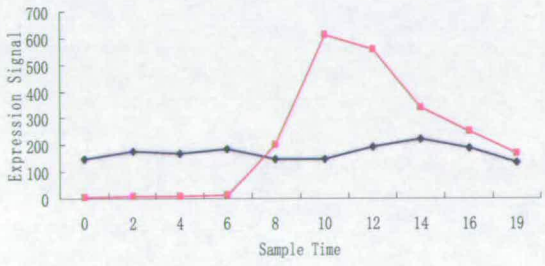
In Figure 6.3, the expression curves of CDK-like and cyclin-like genes in the nineteen pairs are displayed. Murray JAH group (Menges M, *et al*, 2005) synchronized the *Arabidopsis* cells using the drug aphidicolin to block the cells at the G1/S boundary. They then released the cells by removal of the drug and took sequential RNA samples at 2-3 hours intervals over a 19 hour period. Then a transcriptional profiling analysis was carried out to study gene expression during cell cycle progression after aphidicolin treatment using near full genome ATH1 arrays (Data extracted from TAIR database, http://www.arabidopsis.org/). We compared the expression curves of CDK-like and cyclin-like genes in 14 pairs (the other 5 pairs are not displayed because either the CDK or the cyclin's expression data is unavailable).

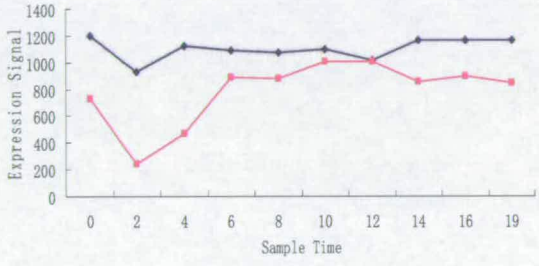


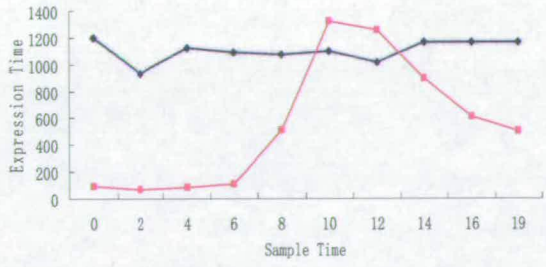


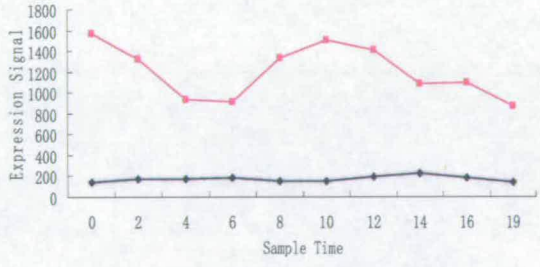












(to be continued)

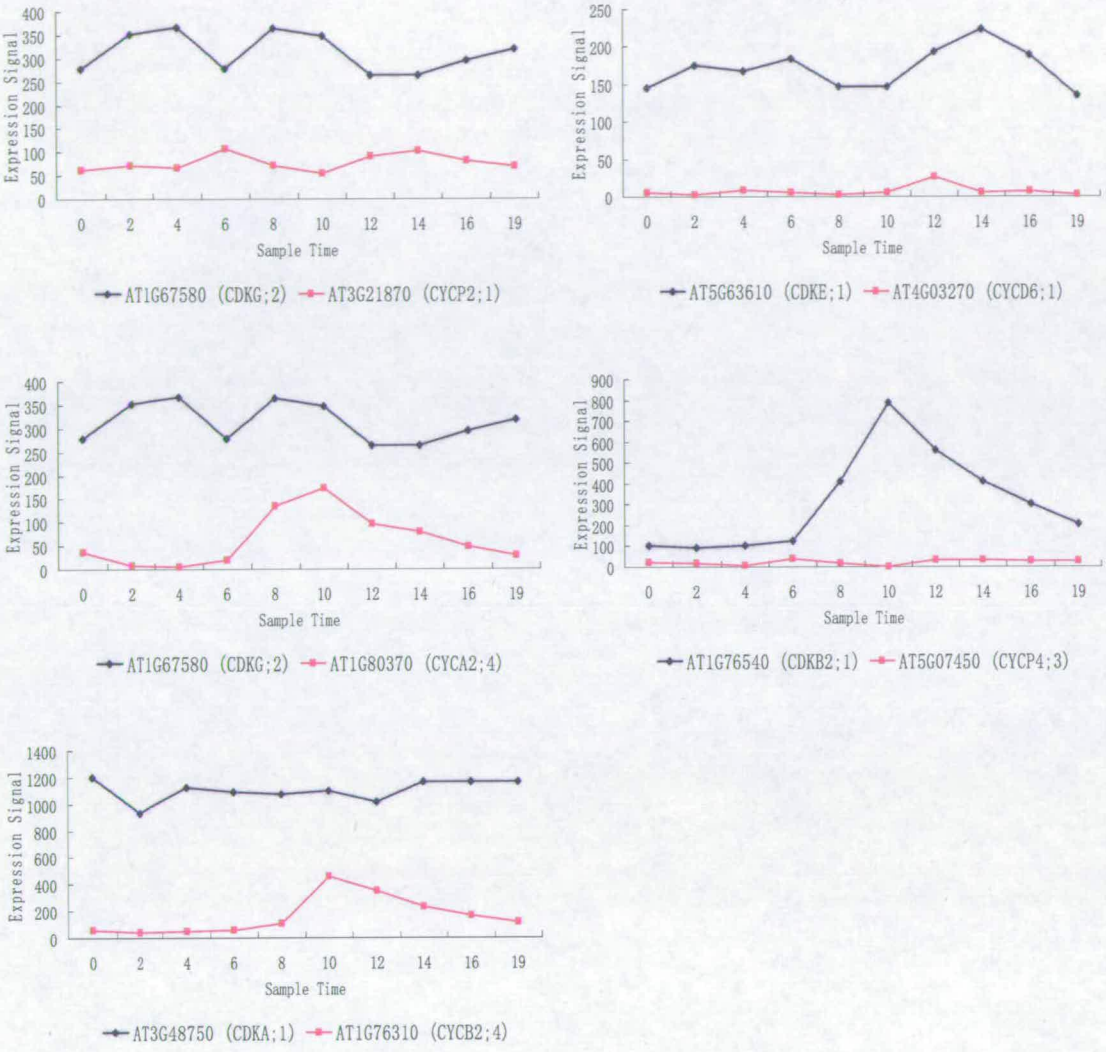


Figure 6.3 The expression level of arabidopsis CDK/cyclin gene after removal of aphidicolin treatment.

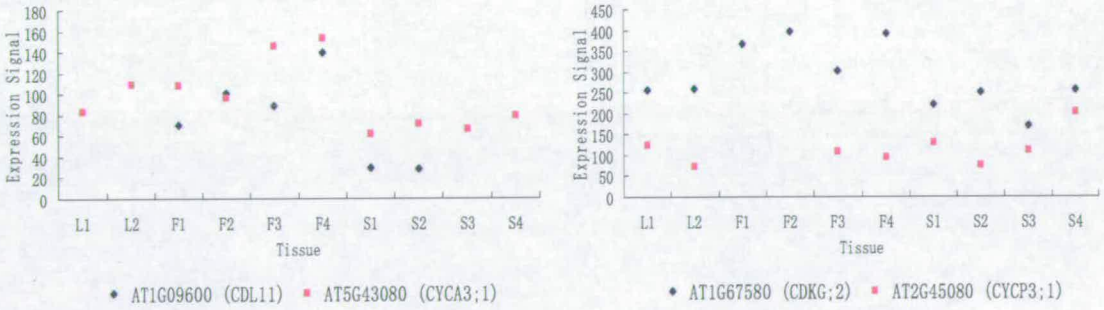
Of the 14 CDK/cyclin like gene pairs, most of the CDK-like genes' expression levels, such as CDKA;1, CDKG;2, and CDKE;1, remain relatively high (micro-array signal larger than 100) through the cell cycle. The mean values of expression signals for CDKA;1, CDKG;2 and CDKE;1 are 1100.7, 312.4, and 170.7; and the standard

deviations of expression signals for CDKA;1, CDKG;2, and CDKE;1 are 81.2, 41.7, and 27.4. The division of standard deviation values by mean values for each gene are then 0.07 for CDKA;1, 0.13 for CDKG;2, and 0.16 for CDKE;1, all not high. Therefore we can regard the expression of these three CDK-like genes as being relatively constant. There are two genes whose expression doesn't fall into this pattern, AT1G09600 (CDL11) and AT1G76540 (CDKB2;1). The expression curve of CDKB2;1 has a sharp peak instead of being smooth as expected. CDL11's expression remains low (expression signal lower than 50 at most phases, and lower than 100 for all the cell cycle) all through the cell cycle. Because of the broad range of expression signal detected by this micro-array experiment, varying from near zero to more than 1000, such low data can be heavily affected by background signal strength during the data normalization and therefore any fluctuation of signal strength at such low level become unreliable.

The expression curves of most cyclins, such as CYCB1;3, CYCB1;4, CYCB2;4, CYCA2;4, CYCP2;1, and CYCD6;1, have obvious peaks, as expected. The expression curve of CYCD3;1 looks a bit abnormal as it has a valley instead of peak. If this abnormal expression curve was not caused by experimental errors, it might indicate that CYCD3;1 may act in the regulation of more than one cell cycle phase entry point, like the human cyclinA (Pagano *et al*, 1992). The expression level of some cyclin-like genes (CYCD6;1, CYCP3;1, and CYCP4;3) remain low (expression signal less than 50) throughout the cell cycle. These unusual facts are either caused by experimental signal detection errors, or the long time intervals of sampling as cyclins might be highly expressed only over a very short time period.

Similar to the expression profiles of CDKs and cyclins at different stages of cell cycle, the CDK and cyclin in an interacting CDK-cyclin pair also need to be co-expressed at least in some tissues. However, if one CDK/cyclin protein is only present in one or two tissue types, the expression of its interacting cyclin/CDK partner should at least not be low. Additionally we need to bear in mind that the expression strength of a gene in a specific tissue is really an average value of the expression strengths of this gene at different phases of cell cycle.

The expression variation of cell cycle core genes in different tissues, for example leaf, flower and stem, were also analyzed by Murrays JAH group (Mengs M *et al*, 2005) (data come from database TAIR, http://www.arabidopsis.org/). Figure 6.4 display the expression profiles of some of our predicted pairs for which expression data are available. The expression level of some pairs, for example CKL11-CYCA3;1, CDKG;2-CYCP3;1, display quite obvious positive correlation. This fact also supports our predictions of these pairs.



(to be continued)

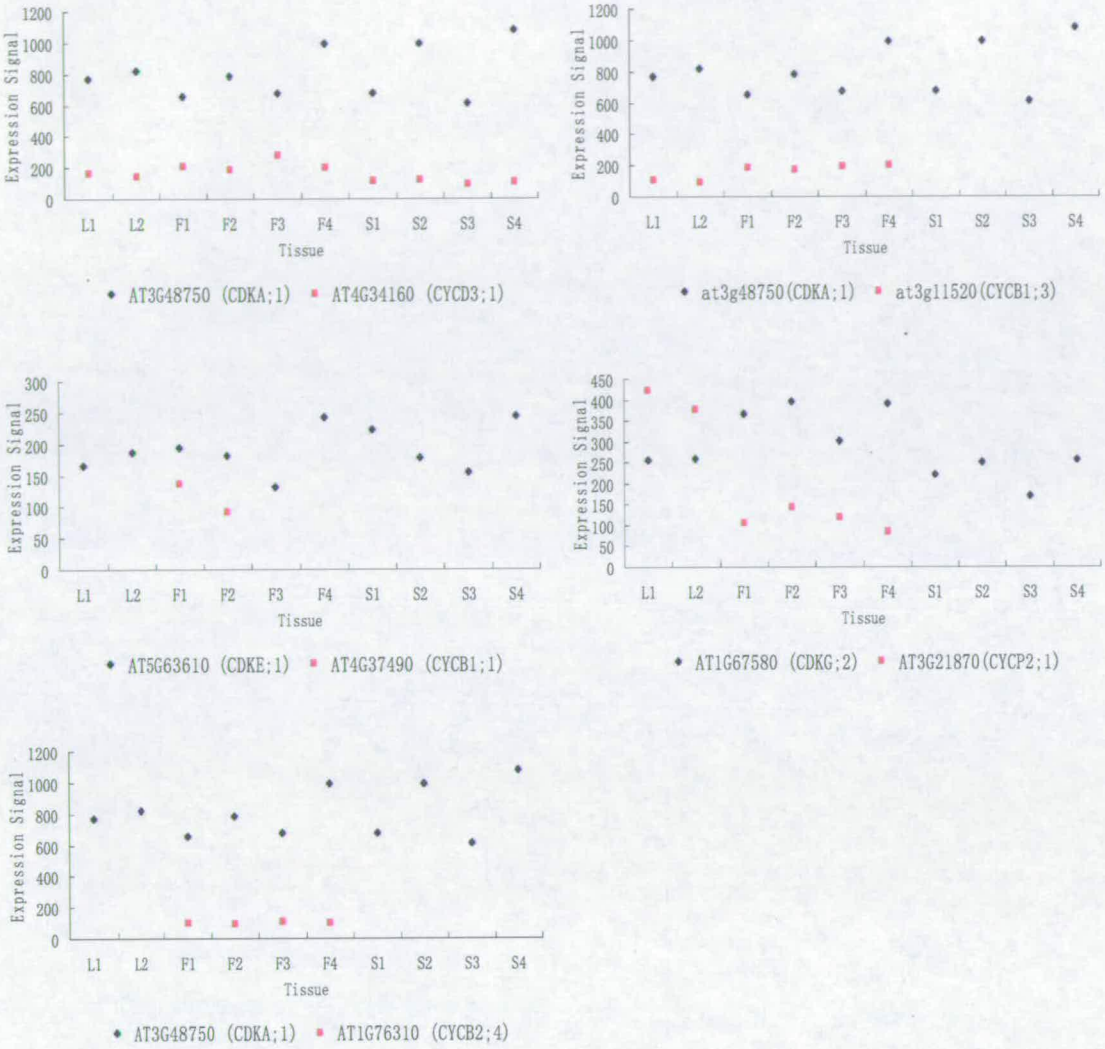


Figure 6.4 The expression profiles of *Arabidopsis* CDK/cyclin genes in different tissues.

L1: Leaf-GH1; L2: Leaf_GH2; F1: Flower-GC5; F2: Flower-GC6; F3: Flower-GH5; S1: Stem_GC7; S2: Stem_GC8; S3: Stem_GH7; S4: Stem_GH8. Samples of the same kind of tissue but with different number are different based on the different Germplasm used (http://www.arabidopsis.org/).

6.3 Future Directions

For the moment, our prediction approach has only been applied to the interaction between CDKs and cyclins. In the future, it is planed to apply this prediction approach to other transient hetero-dimer interactions, even obligate interactions. This approach should be easily transferred to other transient hetero-dimer interaction. As previously outlined, the modelling procedure will be a limiting point of the transferability. The new program 3-D-T-Coffee (O' Sullivan O *et al*, 2004) might help to make application fast and less dependent on human intervention.

The biological weakness of this type prediction approach is of course that the CDK-cyclin pairs predicted are only predicted to have the potential to interact not that they will definitely interact *in vivo*.

7.

SUMMARY

CDKs (cyclin-dependent kinases) are Ser/Thr protein kinases that play an essential role in the regulation of eukaryotic cell proliferation, neuronal and thymus functions, and transcription in animals. Monomeric CDKs are inactive and require both association with a positive regulatory subunit, a cyclin, and phosphorylation of a conserved threonine residue that lies within the activation loop of the CDK, for full activity. The 3-D structures of CDKs and cyclins coming from other species, and/or other subfamilies, can be modelled using known and modelled structures of human CDKs and cyclins as templates. Previous studies and sequence sub-classification of the cyclin and CDK multigene families in the genome of the model plant *Arabidopsis thaliana* have revealed a minimum of 50 cyclin-like, and 35 CDK-like putative gene products. However, the sequence-structure relationships that determine the specificity of protein-protein interactions are not sufficiently understood at present to predict which specific CDK-cyclin pairings are likely to occur, and which are not. By focusing on this specific prediction challenge, we are working towards a better understanding of the biophysical principles governing protein-protein interactions in general, and towards developing computational methodology combining multiple components from several existing methods, that can be applied generally in this field.

The method includes:

ALL-BY-ALL DOCKING: Comparative models for 33 putative CDKs and 35 putative cyclins from *Arabidopsis thaliana* were produced with the program Modeller6, based on

carefully adjusted multiple sequence alignments of each family. The resulting structures were subjected to a large scale molecular docking experiment with ZDOCK in which all CDK-cyclin combinations were considered.

ADDITIONAL SELECTION CRITERIA: Automated protein-protein docking results typically contain too many false positive complexes to be directly useful in practice. We therefore applied two additional criteria to select the best complexes from the ZDOCK result lists, based on:

(i) Orientation - correct relative orientation of CDK and cyclin subunits in the complex;

(ii) Interface – the electrostatic and hydrophobic properties at the interaction surface. Interface electrostatic correlation coefficients (ECC) and hydrophobic correlation coefficients (HCC) were calculated using the program MolSurfer.

CALIBRATION: For calibrating the interface criterion we used a positive, and a negative set of control data. The positive set consisted of 104 non-homologous, transient hetero-dimer complexes. For the negative set 70 "non-complexes" were generated by using ZDOCK to combine proteins that do not normally interact (ZDOCK score > 60; interface size > 600 square angstroms). An optimal cutoff value for CEH (a combination of ECC and HCC) was chosen, based on the interface properties of these two sets.

The results can be divided into two parts, the cross-validation of this approach and the application of this approach to *Arabidopsis thaliana* CDK-cyclin interactions:

• Cross-validation, by randomly selecting 80% of data from each control set to derive the CEH cutoff, consistently yielded separation accuracies around 80% for the other 20% of control complexes. When we applied the entire modelling and interaction

prediction approach to the well-characterized set of human CDKs and cyclins, 80% of the resulting predictions were in agreement with HRPD and Swiss-Prot annotation.

• All-by-all docking between 33 CDK models and 35 cyclin models of *A. thaliana* yielded 83 pairs with outstanding ZDOCK scores (> 60). Of these pairs, 59 are supported by correct orientation between the interacting subunits. Using CEH as an additional selection criterion, we retained 19 most likely interacting CDK-cyclin pairs in *Arabidopsis thaliana*. The most strongly predicted complex is formed between a close homologue of human CDK1/2/3, and a sequence most similar to human cyclinA (human CDK1/2/3-cyclinA are natural pairs). Another predicted complex has recently been confirmed experimentally.

APPENDIX

A. References for Figure 5.14:

CDKs:

B01 AT3G48750
B02 AT1G18040
B03 AT1G66750
B04 AT1G73690
B05 AT1G03740
B06 AT1G09600
B07 AT1G10210
B08 AT1G18670
B09 AT1G20930
B10 AT1G33770
B11 AT1G53050
B12 AT1G54610
B13 AT1G57700
B14 AT1G59580
B15 AT1G67580
B16 AT1G71530
B17 AT1G74330
B18 AT1G76540
B19 AT2G18170
B20 AT2G38620
B21 AT3G05050
B22 AT3G54180
B23 AT3G59790
B24 AT4G10010
B25 AT4G22940
B26 AT4G36450
B27 AT5G10270
B28 AT5G39420
B29 AT5G44290
B30 AT5G50860
B31 AT5G63370
B32 AT5G63610
B33 AT5G64960

Cyclins:

A01 AT2G22490
A02 AT2G45080
A03 AT3G21870

A04 AT3G60550
A05 AT3G63120
A06 AT5G07450
A07 AT5G61650
A08 AT1G15570
A09 AT1G16330
A10 AT1G20610
A11 AT1G44110
A12 AT1G47210
A13 AT1G47220
A14 AT1G47230
A15 AT1G76310
A16 AT1G77390
A17 AT1G80370
A18 AT2G17620
A19 AT2G26760
A20 AT3G11520
A21 AT4G35620
A22 AT4G37490
A23 AT5G06150
A24 AT5G11300
A25 AT5G25380
A26 AT5G43080
A27 Q38818
A28 AT1G70210
A29 AT2G22490
A30 AT3G50070
A31 AT4G03270
A32 AT4G34160
A33 AT4G37630
A34 AT5G02110
A35 AT5G65420
A36 AT5G67260

B Colouring Scheme of ClustalX

(http://bips.u-strasbg.fr/fr/Documentation/ClustalX/#C):

The format of each consensus parameter is:-

c = n% residue_list

where
- c is a character used to identify the parameter.
- n is an integer value used as the percentage cutoff point.
- residue_list is a list of residues denoted by a single character, delimited by a colon (:).

For example: # = 60% w:l:v:i

will assign a consensus character # to any column in the alignment which contains more than 60% of the residues w,l,v and i.

The third section is identified by the header @color, and defines how colors are assigned to each residue in the alignment.

The color parameters can take one of two formats:

1) r = color
2) r = color if consensus_list

where
- r is a character used to denote a residue.
- color is one of the colors in the GDE color lookup table.
- residue_list is a list of residues denoted by a single character, delimited by a colon (:).

Examples: 1) g = ORANGE

will color all glycines ORANGE, regardless of the consensus.

2) w = BLUE if w:%:#

will color BLUE any tryptophan which is found in a column with a consensus of w, % or #.

ClustalX Format Consensus Definition:

@consensus
% = 60% w:l:v:i:m:a:f:c:y:h:p
# = 80% w:l:v:i:m:a:f:c:y:h:p
- = 50% e:d
+ = 60% k:r
g = 50% g
n = 50% n
q = 50% q:e
p = 50% p
t = 50% t:s
A = 85% a
C = 85% c
D = 85% d
E = 85% e
F = 85% f
G = 85% g
H = 85% h
I = 85% i
K = 85% k
L = 85% l
M = 85% m
N = 85% n
P = 85% p
Q = 85% q
R = 85% r
S = 85% s
T = 85% t
V = 85% v
W = 85% w
Y = 85% y

Clustal Format Colour Parameter Definition

g = ORANGE
p = YELLOW
t = GREEN if t:S:T:%:#
s = GREEN if t:S:T:#
n = GREEN if n:N:D
q = GREEN if q:Q:E:+:K:R
w = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
l = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
v = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
i = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p

m = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
a = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p:T:S:s:G
f = BLUE if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
c = BLUE if %:#:A:F:H:I:L:M:V:W:Y:S:P:p
c = PINK if C
h = CYAN if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
y = CYAN if %:#:A:C:F:H:I:L:M:V:W:Y:P:p
e = MAGENTA if -:D:E:q:Q
d = MAGENTA if -:D:E:n:N
k = RED if +:K:R:Q
r = RED if +:K:R:Q

C. Perl Scripts

Script1:

```
#!/usr/bin/perl –w
# Block.pl by Xueping Quan
# read in a alignment file in modeller format, a list file of the non-interface residues of
 # template sequence, and the name of the template

if ($#ARGV !=1) {
  print "This script is to list the residues of sequences in a alignment file according to a
template\n";
  print "If you know that some residues are not in the binding site in the template
structure,\n";
  print "please list their residue numbers in a file with one number a line.\n";
  print "This program read in the alignment in modeller format,\n";
  print "and list the residue numbers of model sequences not in the binding sites.\n";
  print "Useage:\n";
  print "$0 [alignment file] [list file]\n";
  die;
}

$alignname = $ARGV[0];
$listname  = $ARGV[1];

open( LISTFILE, "$listname" ) || die ( "Cannot open file: $!\n" );
@listfile = <LISTFILE>;
chomp @listfile;
close (LISTFILE) || die ( "Cannot close file: $!\n" );

print "Please input the template name: ";
chomp ($temp = <STDIN>);

#divide the alignment file into two part,and then store them into two array.
open (ALIGN, "$alignname") || die "\nCannot open file: $!\n";
@file = <ALIGN>;
close (ALIGN) || die ( "Cannot close file: $!\n" );

$seqno = -1;
$i = 0;

foreach $file (@file) {
  if ($file =~ m/^>P/) {
    ++$seqno;
    $array1[$seqno] = $file;
  }elsif($file =~ m/^sequence/ or $file =~ m/^structure/) {
```

```
    next;
   }else{
    $array3[$seqno][$i] = $file;
    ++$i;
   }
}

# chop the \n of every line  in the two arrays and store them into two new arrays.
@seq = ();
@tem = ();

for ($j = 0; $j < $i; ++$j) {
  $seq[$j] = $array3[0][$j];
}

for ($j = 0; $j <= $#array1; ++$j) {
  if ($array1[$j] =~ m/$temp/) {
    for ($k = 0; $k < $i; ++$k) {
      $tem[$k] = $array3[$j][$k]
    }
  }
}

$seq = join('',@seq);
$tem = join('',@tem);

# remove whitespace and wild card.
$seq =~ s/\s//g;
$tem =~ s/\s//g;

#now explode the strings into an array.This will make it easy to look at each position
@seq = split('',$seq);
@tem = split('',$tem);

# remove the last element '*' in the two arrays.
pop @seq;
pop @tem;

print "$seq\n";
print "\n$tem\n";

#now try to give each residue it's sequence number.
$k = 1;
$j = 0;
```

```
for ($i = 0; $i < @seq; ++$i) {
   $seq2[$i][0] = $seq[$i];
   if ($seq[$i] eq '-') {
     $seq2[$i][1] = 0;
   }else{
     $seq2[$i][1] = $k;
     ++$k;
   }
}

$k = 1;
for ($i = 0; $i < @tem; ++$i) {
    $tem2[$i][0] = $tem[$i];
   if ($tem[$i] eq '-') {
      $tem2[$i][1] = 0;
    }else{
    $tem2[$i][1] = $k;
    ++$k;
   }
}

# now find the residues of template listed on the list file
# and output the corresponding number of residues in the model sequence

@block = ();
$k = 0;

for ($i =0; $i < @tem; ++$i) {
  LINE: for ($j = 0; $j < @listfile; ++$j) {
      if ($tem2[$i][1] == $listfile[$j]) {
        $block[$k] = $seq2[$i][1];
        ++$k;
        last LINE;
      }
  }
}

open(OUTFILE, ">block.lis") or die("Cannot open file: $!\n");

for $numb (@block) {
   print OUTFILE "$numb\n";
}
exit;
```

Script2:

```
#!/usr/bin/perl –w
#PDBcontactauto.pl by Xueping Quan
# Calculate the inter-molecule non-hydrogen atom distance in PDB file

if ($#ARGV !=0) {
   print "The scripts is to calculate the atom distance in PDB file.";
   print "Read in a file that list the pdb file name";
   print " then calculate the sidechain atom distance between different chains";
   print "Useage:\n";
   print "$0 [pdb file list file name]\n";
   die;
}

$pdblist = $ARGV[0];

open ( PDBFILE, "<$pdblist" ) || die ( "Cannot open file: $!\n" );
@pdbs = <PDBFILE>;
chomp @pdbs;
close (PDBFILE) || die ( "Cannot close file: $!\n" );

foreach $pdb (@pdbs) {
  $pdbname = $pdb . ".pdb";
  $conname = $pdb . ".con";

  open ( INFILE, "<$pdbname" ) || die ( "Cannot open file: $!\n" );
  @pdbfile = <INFILE>;
  close (INFILE) || die ( "Cannot close file: $!\n" );


  # Parse the record types of the PDB file
  %transient = parsePDBrecordtypes(@pdbfile);

 # Extract the atoms of all chains in the protein
 # and caluclat the atom distance between residues on different chains
  %result = calculatecontacts($transient{ATOM});

   open( OUTFILE, ">$conname" ) or die( "Cannot open file: $!\n" );

 print OUTFILE "The protein-protein interaction result of $pdbname:\n";
 print OUTFILE "numb cha1 res1 reID1 atom1 distan cha2 res2 reID2 atom2\n";

 @k = sort {$a <=> $b} (keys %result);
```

```
 foreach $k (@k) {
  $v  = $result{$k};
  $k1 = sprintf("%4d", $k);
  print OUTFILE "$k1 $v\n";
 }

 close ( OUTFILE ) or die ( "Can not close file: $!\n" );
}

exit;


############################
#Subroutines
############################


# parsePDBrecordtypes
# given an array of a PDB file, return a hash
# with keys = record type names
# values = scalar containing lines for that record type

sub parsePDBrecordtypes {

    use strict;

    my @file = @_;

    my %recordtypes = ();

    foreach my $line (@file) {
    # Get the record type name which begins at the start
    # of the line and ends at the first space
    # The pattern (\S+) is returned and saved in $recordtype
    my ($recordtype) = ($line =~ /^(\S+)/);

    # .= fails if a key is undefined, so we have to
    # test for definition and use either .= or = depending
    if(defined $recordtypes{$recordtype} ) {
       $recordtypes{$recordtype} .= $line;
    }else{
        $recordtypes{$recordtype} = $line;
    }
   }
   return %recordtypes;
```

```
}


# calculatecontacts
# extract x, y, and z corrdinates, serial number and element symbol from PDB ATOM
record type
# get a hash with key=serial number, value=corrdinates in a string
# calculate the atom distance between different residues on different chains

sub calculatecontacts {

  my($atomrecord) = @_;

  use strict;
  use warnings;

   # Set the array of atoms to empty
  my @atoms = ();
  my $i = 0;

  # Turn the scalar into an array of ATOM lines
  my(@atomrecord) = split(/\n/, $atomrecord);

  foreach my $record (@atomrecord) {
    my $number  = substr($record,  6, 5); #columns 7-11
    my $residu  = substr($record, 17, 3); #columns 18-20
    my $residue = sprintf("%4s", $residu);
    my $chain   = substr($record, 21, 1); #columns 22
    my $reID    = substr($record, 22, 4); #columns 23-26
    my $resID   = sprintf("%5d", $reID);
    my $x       = substr($record, 30, 8); #columns 31-38
    my $y       = substr($record, 38, 8); #columns 39-46
    my $z       = substr($record, 46, 8); #columns 47-54
    my $atom    = substr($record, 12, 4); #columns 77-78


    # $number, $chain and $element may have leading spaces: strip them
     $number  =~ s/^\s*//;

    # $atom have gear spaces to substitute
     $atom  =~ s/\s*//g;

    # Store information in a two-dimensional array
    my $j = 0;
```

```
  $atoms[$i][$j]   = "$number";
  $atoms[$i][++$j] = "$residue";
  $atoms[$i][++$j] = "$resID";
  $atoms[$i][++$j] = "$chain";
  $atoms[$i][++$j] = "$x";
  $atoms[$i][++$j] = "$y";
  $atoms[$i][++$j] = "$z";
  $atoms[$i][++$j] = "$atom";
  ++$i;
  }

 my $chID = $atoms[0][3];

# begin calculate
my %contacts = ();

# use the number of the keys of @atoms as a control number

my $n = 0;

for(my $k = 0; $k < $i; ++$k ) {

   my $num1   = $atoms[$k][0];
   my $res1   = $atoms[$k][1];
   my $reID1  = $atoms[$k][2];
   my $chai1  = $atoms[$k][3];
   my $x1     = $atoms[$k][4];
   my $y1     = $atoms[$k][5];
   my $z1     = $atoms[$k][6];
   my $ato1   = $atoms[$k][7];

   for(my $m = 0; $m < $i; ++$m ) {

     my $num2   = $atoms[$m][0];
     my $res2   = $atoms[$m][1];
     my $reID2  = $atoms[$m][2];
     my $chai2  = $atoms[$m][3];
     my $x2     = $atoms[$m][4];
     my $y2     = $atoms[$m][5];
     my $z2     = $atoms[$m][6];
     my $ato2   = $atoms[$m][7];

      if ( $chai1 eq $chID && $chai2 ne $chID ) {
unless ($ato1 =~ /^H/ && $ato2 =~ /^H/) {
    my $dis1 = sqrt(($x1-$x2)**2 + ($y1-$y2)**2 + ($z1-$z2)**2);
```

```
    my $dis  = sprintf("%6.3f", $dis1);
      if( $dis <= 5.00 ){
        my $cha1   =sprintf("%4s", $chai1);
        my $cha2   = sprintf("%4s", $chai2);
        my $atom1  = sprintf("%5s", $ato1);
        my $atom2  = sprintf("%5s", $ato2);

$contacts{$n} = "$cha1 $res1 $reID1 $atom1 $dis $cha2 $res2 $reID2 $atom2";
          ++$n;

          }
      }
   }
  }
 }
   # Return the result
   return %contacts;
}
```

Script3:

```
#!/usr/bin/perl –w
#contactresiduepair.pl by Xueping Quan
# read in the output file of PDBcontactauto.pl
# list the inter-chain contacting residue pairs in a complex

use strict;
use warnings;

# First tell the user to input the path of their files
print "Please input the path of you contact file: ";
chomp( my $filedir = <STDIN> );

opendir(DIR, $filedir ) || die ( "\nCannot open $filedir: $!\n" );
readdir(DIR);
closedir DIR;

my @CONfile = ();

print "Please input your contact file name: ";
chomp ( my $confilename = <STDIN> );
```

```
# Now we open the file and associate a filehandle with it.
open( CONFILE, "$confilename" ) || die ( "Cannot open file: $!\n" );

#Now we actually reading the file
@CONfile = <CONFILE>;

close (CONFILE) || die( "Cannot close file : $!\n" );


# list all the residue pairs that contact each other

 my @table = makelist ( @CONfile );

# analyse the residue pairs and give the numbers of different residue pairs
 my ($total,@graph) = analysepairs ( @table );


# Write the results to a file called "pairs.lis"
if ( -e 'pairs.lis' ) {
  print( "Do you want to write over pairs.lis? (yes or no): ");
  chomp ( my $response = <STDIN> );
  rename( 'pairs.lis', 'pairs.old' )
    or die( "Error renaming : $!" )
      if ( $response eq 'no' );
}

open( OUTFILE, ">pairs.lis" ) or die( "Cannot open file: $!\n" );

print OUTFILE "This is the numbers of different residue pairs of $confilename:\n";

my $bref;
  for $bref (@graph) {

  print OUTFILE " @$bref\n";

  }

print OUTFILE "Overall there are $total amino acid pairs in this interface!\n";
close ( OUTFILE ) or die ( "Can not close file: $!\n" );

exit;


##########################
#Subroutines
```

```
##########################

# make list of residue pairs
sub makelist {

  my @file = @_;

  use strict;
  use warnings;

 #read the file into an two-directional array.
  my $i = 0;
  my $m = 0;
  my @cont = ();
  foreach my $line (@file) {
    if ($i > 1) {
      my $resID1 = substr($line, 15, 5); #column 16-20
      my $res1   = substr($line, 11, 3); #column 12-14
      my $res2   = substr($line, 39, 4); #column 40-44
      my $resID2 = substr($line, 44, 5); #colimn 44-49

      # put $resID2 and $res2 into one scalar
      my $residue2 = $resID2 . $res2;

      # store information in an array
      $cont[$m][0] = $resID1;
      $cont[$m][1] = $res1;
      $cont[$m][2] = $residue2;
       ++$m;
     }else{
     ++$i;
     }
  }

  # extract the number of residues on chain 2 that any residues
  # on chain 1 are in contact with
  # put the results into a two-directional array

   my @values = ();
   my $plus   = 0;
   my $total  = 0;
   my @temp   = ();
   my $k      = 0;
   my $j      = 0;
   my @string;
```

```
$m = $#cont;
for ( my $w = 0; $w < @cont; ++$w ) {

   my $ID1   = $cont[$w][0];
   my $resd1 = $cont[$w][1];
   my $resd2 = $cont[$w][2];
   my $w1    = $w - 1;
   if ( $w == 0 ) {
    $values[$k][0] = $ID1;
    $values[$k][1] = $resd1;
    $temp[$j]      = $resd2;
    ++$j;
   } elsif ($w >= 1 && $w < $m) {
     if ($ID1 == $values[$k][0]) {
        $temp[$j] = $resd2;
        ++$j;
      } else {
       @string = processtemp(@temp);
       my $t = 0;
       foreach(@string){
       $values[$k][2] = $string[$t];
       ++$t;
       ++$k;
       $values[$k][0] = $cont[$w1][0];
       $values[$k][1] = $cont[$w1][1];
       }
       $values[$k][0] = $ID1;
       $values[$k][1] = $resd1;
       @temp = ();
       $j = 0;
       $temp[$j] = $resd2;;
       ++$j;
       }
     }elsif ($w == $m) {
       if ($ID1 == $values[$k][0]){
          $temp[$j] = $resd2;
          @string = processtemp(@temp);
          my $t2 = 0;
          my $ind = $#string;
          foreach(@string){
           if($t2 <  $ind) {
            $values[$k][2] = $string[$t2];
            ++$t2;
            ++$k;
```

```
            $values[$k][0] = $ID1;
            $values[$k][1] = $resd1;
           }elsif($t2 == $ind) {
            $values[$k][2] = $string[$t2];
           }
         }
        }else{
            @string = processtemp(@temp);
            my $t3 = 0;
            foreach(@string){
            $values[$k][2] = $string[$t3];
            ++$k;
            ++$t3;
            $values[$k][0] = $cont[$w1][0];
            $values[$k][1] = $cont[$w1][1];
            }
            $values[$k][0] = $ID1;
            $values[$k][1] = $resd1;
            $values[$k][2] = $resd2;
         }
       }

      }

    return (@values);
  }

# processtemp
# calculate every part of the contacts result that
# have the same residue ID on chainA
sub processtemp{
  my @temp2 = @_;

  use strict;
  use warnings;

  my $value = 1;

  my @temp = sort @temp2;
  my $index = scalar (@temp2);
  my $residuelist;
  my @list = ();
  my $m = 0;

  for (my $i = 0; $i < $index; ++$i) {
```

```
    my $name = $temp[$i];

    if ($i == 0) {
      $list[$m] = $name;
    }elsif($i > 0 && $i < $index ) {
      if ($name eq $list[$m]) {
        next;
      }else{
       ++$m;
       $list[$m] = $name;
      }
    }
   }
  return @list;
}

#analysepairs
#analyse the contacting residue pairs and calculate the numbers of defferent
#residue pairs
sub analysepairs{
  my @accept = @_;

  use strict;
  use warnings;

  my $m = 0;
  my @string = ();

  foreach (@accept) {
    my $res1  = $accept[$m][1];
    my $res2  = $accept[$m][2];

  # $resd2 have leading spaces and numbers, strip them.
    $res2 =~ s/^\s*\d*//;

    my $pair = $res1 . $res2;

    $string[$m] = $pair;
    ++$m;
  }

  my @result = ();
  my $i = 0;
  my @pairs = sort @string;
  my $k = $#pairs;
```

```
my $value = 1;
my $value1;

for ( $m = 0; $m < @pairs; ++$m ) {
 if ($m == 0) {
 $result[$i][0] = $pairs[$m];
 }elsif ($m > 0 && $m < $k) {
   if ($pairs[$m] eq $result[$i][0]) {
     ++$value;
   }else{
     $value1 = sprintf ("%3d", $value);
     $result[$i][1] = $value1;
     $value = 1;
     ++$i;
     $result[$i][0] = $pairs[$m];
   }
 }else {
   if ($pairs[$m] eq $result[$i][0]) {
       ++$value;
      $value1 = sprintf ("%3d", $value);
      $result[$i][1] = $value1;
   }else {
     $value1 = sprintf ("%3d", $value);
     $result[$i][1] = $value1;
     ++$i;
     $result[$i][0] = $pairs[$m];
     $value = 1;
     $value1 = sprintf ("%3d", $value);
     $result[$i][1] = $value1;
   }
 }
}

 my $number = 0;
 for ( $m = 0; $m < @result; ++$m ) {
    $number += $result[$m][1];
 }
 return ($number,@result);
}
```

Script4:

```
#!/usr/bin/perl -w
#All-by-all-docking-automate.pl by Xueping Quan
# this script is automate the the process of large scale docking
# the pdb file names which will be docking are stored in two file:
```

```
# CDKs and cyclins.
# scripts read in the two file, store the two set of names into two arrays,
# then run all by all docking.

#read in the two files

if ($#ARGV !=1) {
   print "The scripts is to automate the process of large scale docking.";
   print "Read in two files that list the pdb file name of Receptors and ligands";
   print " then start zdock for docking.";
   print "Useage:\n";
   print "$0 [receptor list file name] [ligand list file name]\n";
   die;
}

$receptorlist = $ARGV[0];
$ligandlist = $ARGV[1];

open ( RECEPTOR, "<$receptorlist" ) || die ( "Cannot open file: $!\n" );
@receptors = <RECEPTOR>;
chomp @receptors;
close (RECEPTOR) || die ( "Cannot close file: $!\n" );

open ( LIGAND, "<$ligandlist" ) || die ( "Cannot open file: $!\n");
@ligands = <LIGAND>;
chomp @ligands;
close (LIGAND) || die ( "Cannot close file: $!\n" );

open ( OUTFILE, ">zdockrest.log") || die ( "Cannot open file: $!\n");

for ($i = 0; $i < @receptors; ++$i) {
  $CDK = $receptors[$i];
  @outCDK = split(",$CDK);
  splice(@outCDK,-10);
  $CDKname = join(",@outCDK);
  for ($j = 0; $j < @ligands; ++$j) {
    $cyclin = $ligands[$j];
    @outcyclin = split(",$cyclin);
    splice(@outcyclin,-10);
    $cyclinname = join(",@outcyclin);

    $output = $CDKname.$cyclinname;
    @args = ("./timeout.pl", "7h", "./zdock -R $CDK -L $cyclin -o $output");
    system (@args) == 0 or die "system @args failed: $?\n";
    print OUTFILE "Pair $CDK and $cyclin have been docked!\n";
```

```
  }
}

close ( OUTFILE ) or die ( "Can not close file: $!\n" );
exit;
```

Script5:

```
#!/usr/bin/perl -w
#
# $Id: timeout.pl,v 1.9 2002/12/10 02:53:38 jmates Exp $
#
# Copyright (c) 2002, Jeremy A. Mates.  This script is free software;
# you can redistribute it and/or modify it under the same terms as
# Perl itself.
#
# Run perldoc(1) on this file for additional documentation.
#
#############################################################################
#
# REQUIREMENTS

require 5;

use strict;

#############################################################################
#
# MODULES

use Carp;       # better error reporting
use Getopt::Std;  # command line option processing

#############################################################################
#
# VARIABLES

my $VERSION;
($VERSION = '$Revision: 1.9 $ ') =~ s/[^0-9.]//g;

my (%opts, %features, $t0);
```

```
# how to convert short human durations into seconds
my %factor = (
 'w' => 604800,
 'd' => 86400,
 'h' => 3600,
 'm' => 60,
 's' => 1,
);

#####################################################################
#
# MAIN

# optional high resolution timers
eval { require Time::HiRes; };
unless ($@) {
 require Time::HiRes;
 $features{'Time::HiRes'} = 1;
}

# parse command-line options
getopts('h?v', \%opts);

help() if exists $opts{'h'} or exists $opts{'?'};

if (exists $opts{'v'}) {
 $features{'verbose'} = 1;
}

# regular program arguments
my $duration = shift;
help() unless @ARGV;

# figure out duration, start timer, and fork/exec to run program
my $timeout = duration2seconds($duration);
$duration = seconds2duration($timeout);

$t0 = [Time::HiRes::gettimeofday()] if $features{'Time::HiRes'} and
$features{'verbose'};

my $pid = open WATCH, "-|";

if ($pid) { # parent
 eval {
```

```
  local $SIG{ALRM} = sub { die "alarm\n" };

  alarm $timeout;

  # ergh, need STDERR output pass through... Expect??
  while (<WATCH>) {

   # keep track of output frequency?
   print;
  }
  close WATCH or warn "Warning: kid exited $?\n";

  # so one knows how long positive runs take
  warn "Info: program ran for ",
   sprintf("%.1f", Time::HiRes::tv_interval($t0)), " seconds\n"
   if $features{'Time::HiRes'} and $features{'verbose'};

  alarm 0;
 };
 if ($@) {
  die unless $@ eq "alarm\n";

  warn "Error: timeout ($duration) exceeded: killing pid $pid\n";

  for my $signal (qw(TERM INT HUP KILL)) {
   last if kill $signal, $pid;
   sleep 2;
   warn "Warning: kill of $pid (via $signal) failed...\n";
  }

 }
} else {  # child
 exec @ARGV or die "Error: could not exec: $!\n";
}

##############################################################################
#
# SUBROUTINES

# takes duration such as "2m3s" and returns number of seconds.
sub duration2seconds {
 my $tmpdur = shift;
 my $timeout;

 # assume raw seconds for plain number
```

```
 if ($tmpdur =~ m/^\d+$/) {
  $timeout = $tmpdur * 60;

 } elsif ($tmpdur =~ m/^[wdhms\d\s]+$/) {

  # match "2m 5s" style input and convert to seconds
  while ($tmpdur =~ m/(\d+)\s*([wdhms])/g) {
   $timeout += $1 * $factor{$2};
  }
 } else {
  die "Error: unknown characters in duration.\n";
 }

 unless (defined $timeout and $timeout =~ m/^\d+$/) {
  die "Error: unable to parse duration.\n";
 }

 return $timeout;
}

# takes seconds and returns a shorthand duration format.
sub seconds2duration {
 my $tmpsec = shift;

 unless (defined $tmpsec and $tmpsec =~ m/^\d+$/) {
  die "Error: argument not an integer";
 }

 my $seconds = $tmpsec % 60;
 $tmpsec = ($tmpsec - $seconds) / 60;
 my $minutes = $tmpsec % 60;
 $tmpsec = ($tmpsec - $minutes) / 60;

 #  my $hours = $tmpsec;
 my $hours = $tmpsec % 24;
 $tmpsec = ($tmpsec - $hours) / 24;
 my $days  = $tmpsec % 7;
 my $weeks = ($tmpsec - $days) / 7;

 # TODO better way to do this?
 my $temp = ($weeks) ? "${weeks}w" : '';
 $temp .= ($days)   ? "${days}d"   : '';
 $temp .= ($hours)  ? "${hours}h"  : '';
 $temp .= ($minutes) ? "${minutes}m" : '';
 $temp .= ($seconds) ? "${seconds}s" : '';
```

```
 return $temp;
}

# a generic help blarb
sub help {
 print <<"HELP";
Usage: $0 duration program [program args]

Stops operation of long running programs.  Duration is either seconds,
or a shorthand format of "2m3s" for 123 seconds.

Options for version $VERSION:
 -h/-?  Display this message

 -v    Verbose.  Prints program run time unless timeout is hit.

Run perldoc(1) on this script for additional documentation.

HELP
 exit;
}

##############################################################################
#
# DOCUMENTATION

=head1 NAME

timeout.pl - stop operation of long running programs

=head1 SYNOPSIS

Break out of sleep program after five seconds:

  $ timeout.pl 5s sleep 60

=head1 DESCRIPTION

=head2 Overview

This script allows programs to be stopped after a specified period of
time.  Practical uses for this script include escape from buggy
programs that stall from Makefile, where a SIGINT to stop the program
will also stop make.
```

=head2 Normal Usage

 $ timeout.pl duration program [program args]

See L<"OPTIONS"> for details on the command line switches supported.

The duration can either be a number (raw seconds), or a shorthand format of the form "2m3s" for 120 seconds. The following factors are recognized:

 w - weeks
 d - days
 h - hours
 m - minutes
 s - seconds

Multiple factors will be added together, allowing easy addition of time values to existing timeouts:

 $ timeout.pl 3s3s sleep 60

Would only allow the sleep to run for six seconds.

An error will occur if the script is unable to parse the supplied duration.

=head1 OPTIONS

This script currently supports the following command line switches:

=over 4

=item B<-h>, B<-?>

Prints a brief usage note about the script.

=item B<-v>

Verbose mode. Currently prints program run time unless the timeout is reached.

=back

=head1 BUGS

=head2 Reporting Bugs

Newer versions of this script may be available from:

http://sial.org/code/perl/

If the bug is in the latest version, send a report to the author.
Patches that fix problems or add new features are welcome.

=head2 Known Issues

No known bugs.

=head1 TODO

Currently, a hard upper time limit must be specified. In theory, one could watch the output from the program and stop the program if it remains idle for some period of time.

Since using a piped read from a program, likely cannot supply STDIN to the program in question.

Make sure signals are properly passed back from command being run and reported on?

=head1 SEE ALSO

perl(1), perlipc(1)

=head1 AUTHOR

Jeremy A. Mates, http://sial.org/contact/

=head1 COPYRIGHT

Copyright (c) 2002, Jeremy A. Mates. This script is free software; you can redistribute it and/or modify it under the same terms as Perl itself.

=head1 VERSION

$Id: timeout.pl,v 1.9 2002/12/10 02:53:38 jmates Exp $

=cut

Script6:

```
#!/usr/bin/perl -w
#zdockoutfileedit.pl by Xueping Quan

if ($#ARGV != 0) {
   print " This script is to divide zdock complex pdb files into 2 files";
   print "each containing one chain".
   print "Useage:\n";
   print "$0 [pdb file list]\n";
   die;
}

$pdblist = $ARGV[0];

open (PDBLIST, "<$pdblist" ) || die ( "Cannot open file:$!\n" );
@pdbs = <PDBLIST>;
chomp @pdbs;
close (PDBLIST) || die ( "Cannot close file:$!\n" );

foreach $pdbID (@pdbs) {

open (PDB, "<$pdbID") || die ("Cannot open file:$!\n");
@pdb = <PDB>;
close (PDB) || die ("Cannot close file:$!\n");
$i = 0;
$j = 0;
@output = split(',$pdbID);
splice(@output, -4);
$entry = join(', @output);
@chain = ();

 for ($m = 0; $m < @pdb; ++$m) {
  $line = $pdb[$m];
  $title = substr($line, 0, 4);
  if ($title eq 'ATOM') {
   $chain[$i][$j] = $line;
   ++$j;
  }elsif($title eq 'TER ') {
   ++$i;
   $j = 0;
  }
 }

 for ($m = 1; $m < 3; ++$m) {
```

```
   $newchain = $chain[$m];
   $filename = $entry.'_'.$m.'.pdb';
   open (OUTFILE, ">$filename") || die ("Cannot open file:$!\n");

   for ($n = 0; $n < @$newchain; ++$n) {
  print OUTFILE "$chain[$m][$n]";
    }
   close (OUTFILE) || die ("Cannot close file:$!\n");

}
}
```

Scripts7:

```
#!/usr/bin/perl –w
#pdb2pqrautomate.pl by Xueping Quan
# this script is automate the the process of large scale pdb2pqr
# the pdb file names whose charges will be assigned
# is stored in a list file
# scripts read in the list file, store into a array, then run apbs.

#read in the pdb list files

if ($#ARGV !=0) {
   print "The scripts is to automate the process of large scale pdb2pqr.";
   print "Read in the file that list the pdb file name";
   print " then start pdb2pqr.";
   print "Useage:\n";
   print "$0 [pdb file list file name] \n";
   die;
}

$pdblist = $ARGV[0];

open ( PDBFILE, "<$pdblist" ) || die ( "Cannot open file: $!\n" );
@pdbs = <PDBFILE>;
chomp @pdbs;
close (PDBFILE) || die ( "Cannot close file: $!\n" );

open ( OUTFILE, ">pdb2pqrs.log") || die ( "Cannot open file: $!\n");

for ($i = 0; $i < @pdbs; ++$i) {
  $pdb = $pdbs[$i];
  @outpdb = split(",$pdb);
  splice(@outpdb,-3);
  $pdbname = join(",@outpdb);
```

```
  $pqrname = $pdbname.'pqr';
 @args = ("./pdb2pqr.py", "--ff=amber", "$pdb", "//home/xueping/pdb2pqr-
0.1.0/$pqrname" );
 system (@args) == 0 or die "system @args failed: $?\n";
 print OUTFILE "The pqr file of $pdb has been created!\n";
}
close (OUTFILE) or die ("Cannot close file: $!\n");

exit;
```

Script8:

```
#!/usr/bin/perl –w
#apbsautomate.pl by Xueping Quan
# this script is automate the the process of large scale apbs
# the pqr file names whose electrostatic potential will be calculated
# is stored in a list file
# scripts read in the list file, store into an array, then run apbs.

#read in the pqr list files

if ($#ARGV !=1) {
   print "The scripts is to automate the process of large scale apbs.";
   print "Read in the file that list the pqr file name and the apbs  input file";
   print " then start apbs.";
   print "Useage:\n";
   print "$0 [pqr file list file name] [apbs input file]\n";
   die;
}

$cgilist = $ARGV[0];
$input = $ARGV[1];

open ( CGIFILE, "<$cgilist" ) || die ( "Cannot open file: $!\n" );
@cgis = <CGIFILE>;
chomp @cgis;
close (CGIFILE) || die ( "Cannot close file: $!\n" );

open ( INFILE, "<$input" ) || die ( "Cannot open file: $!\n" );
@apbsinp = <INFILE>;
close (INFILE) || die ( "Cannot close file: $!\n" );

open ( OUTFILE, ">apbs.log") || die ( "Cannot open file: $!\n");
```

```
@newinp = ();
for ($i = 0; $i < @cgis; ++$i) {
  $pqr = $cgis[$i];
  @outpqr = split(",$pqr);
  splice(@outpqr,-4);
  $pqrname = join(",@outpqr);

    $j = 0;
    foreach $line (@apbsinp) {
    @parts = split(' ', $line);
    $type = $parts[0];

      if ($type eq 'read') {
        $oldcgi = $parts[3];
        $line =~ s/$oldcgi/$pqr/;
        $newinp[$j] = $line;
        ++$j;
      }elsif($type eq 'elec'){
       $oldcgi = $parts[2];
       $line =~ s/$oldcgi/$pqrname/;
       $newinp[$j] = $line;
       ++$j;
       print "$j\n";
      }elsif($type eq 'write') {
       $oldcgi = $parts[3];
       $line =~ s/$oldcgi/$pqrname/;
       $newinp[$j] = $line;
       ++$j;
      }else{
       $newinp[$j] = $line;
       ++$j;
      }
     }
  open (INPFILE, ">$input") || die ( "Cannot open file: $!\n" );

   foreach $line  (@newinp) {
   print INPFILE "$line";
   }
   close (INPFILE) or die ("Cannot close file: $!\n");

  @args = ("./apbs", "$input");
  system (@args) == 0 or die "system @args failed: $?\n";
  print OUTFILE "The grd file of $pqr has been created!\n";
}
close (OUTFILE) or die ("Cannot close file: $!\n");
```

```
exit;
```

Script9:

```
#!/usr/sbin/perl –w
#molsurferautomate.pl by Xueping Quan
# this script is automate the the process of large scale molsurfer
#before run this script, set the CLASSPATH to the directory that contains the MolSurfer
#class files

if ($#ARGV !=0) {
   print "The scripts is to automate the process of large scale molsurfer.";
   print "Read in the file that list the pdb entry names to be calculated.";
   print "then start molsurfer.";
   print "Useage:\n";
   print "$0 [pdb entry name list file name]\n";
   die;
}

$pdblist = $ARGV[0];

open ( PDBFILE, "<$pdblist" ) || die ( "Cannot open file: $!\n" );
@pdbs = <PDBFILE>;
chomp @pdbs;
close (PDBFILE) || die ( "Cannot close file: $!\n" );

open ( OUTFILE, ">molsurfer.log") || die ( "Cannot open file: $!\n");

@args = ();
@parts = ();
@parts1 = ();
@parts2 = ();
@parts3 = ();

for ($i = 0; $i < @pdbs; ++$i) {
   $line = $pdbs[$i];
   @parts = split(' ', $line);
   $pdbY = $parts[0];
   $pdbA = $parts[1];
   $pdbB = $parts[2];
   @parts1 = split('', $pdbY);
   @parts2 = split('', $pdbA);
```

```
@parts3 = split(', $pdbB);
splice(@parts1, -4);
splice(@parts2, -4);
splice(@parts3, -4);
$pdb  = join(', @parts1);
$pdbC = join(', @parts2);
$pdbD = join(', @parts3);
$grd1 = $pdbC . ".grd";
$grd2 = $pdbD . ".grd";
$outfile = $pdb . ".ambermol";

@args = ("cp", "$pdbA", "pdb1.pdb");
system (@args) == 0 or die "system @args failed: $?\n";
@args = ("cp", "$pdbB", "pdb2.pdb");
system (@args) == 0 or die "system @args failed: $?\n";
@args = ("cp", "$pdbY", "Y.pdb");
system (@args) == 0 or die "system @args failed: $?\n";
@args = ("cp", "$grd1", "grd1.grd");
system (@args) == 0 or die "system @args failed: $?\n";
@args = ("cp", "$grd2", "grd2.grd");
system (@args) == 0 or die "system @args failed: $?\n";

@args = ("./adsi.exe");
system (@args) == 0 or die "system @args failed: $?\n";
@args = ("./adsiep4apbs");
system (@args) == 0 or die "system @args failed: $?\n";
@args = ("mv", "adsiep.interface", "Y.interface");
system (@args) == 0 or die "system @args failed: $?\n";

@args = ("timeout.pl", "30s", "/usr/java/bin/java IMap Y /usr/java/lib:.");
system (@args) == 0 or die "system @args failed: $?\n";

@args = ("mv", "molsurfer.txt", "$outfile");
system (@args) == 0 or die "system @args failed: $?\n";

print OUTFILE "The molsurfer file of $pdb has been created!\n";
}
close (OUTFILE) or die ("Cannot close file: $!\n");

exit;
```

Script10:

```
#!/usr/bin/perl -w
#ZDOCK-zscore-matrix.pl by Xueping Quan
```

```
if ($#ARGV !=0 ) {
   print "The scripts is to caluclate the z scores of a set of zdock value stored in a list of
files.";
   print "Read in a file that list the file names that need to read.";
   print "These files contain a CDK name and a list of cyclins and corresponding value";
   print "all these files should be in the same cyclin list";
   print "the output file will contain a matrix of all the ZDOCK score file\n";
   print "and a matrix of all the zscore file\n";
   print "Useage:\n";
   print "$0 [list file name]\n";
   die;
}

$listfile = $ARGV[0];

open ( FILE, "<$listfile" ) || die ( "Cannot open file: $!\n" );
@lists = <FILE>;
chomp @lists;
close (FILE) || die ( "Cannot close file: $!\n" );

print "Please input the path of yours files: ";
chomp ( $path = <STDIN> );
$l = 0;
@outputs = ();

foreach $filename (@lists) {
  opendir( DIR, $path ) || die ( "\nCannot open directory: $!\n" );
  readdir(DIR);
  closedir(DIR);

  open (SOURCEFILE, "<$filename" ) || die ( "\nCannot open file: $!\n" );

  @inputs = <SOURCEFILE>;


 LINE:  for ($i = 0; $i < @inputs; ++$i) {
    $j = $i - 1;
    if ($i == 0) {
     next LINE;
    }else{
     $cyclin = $inputs[$i];
     $value = substr($cyclin, 10, 5);
     chomp $value;
     $outputs[$j][$l] = $value;
```

```
    }
  }
  ++$l;
  print "$l\n";
}

$sum = 0;

for ($i = 0; $i < @outputs; ++$i) {
  for ($j = 0; $j < $l; ++$j) {
   $sum += $outputs[$i][$j];
  }
}

$n = 0;
for ($i = 0; $i < @outputs; ++$i) {
 for ($j = 0; $j < $l; ++$j) {
   $value = $outputs[$i][$j];
   if ($value !=0 ) {
    ++$n;
   }
 }
}


$mean = $sum / $n;
$medium = 0;

for ($i = 0; $i < @outputs; ++$i) {
  for ($j = 0; $j < $l; ++$j) {
   $value = $outputs[$i][$j];
   if ($value != 0) {
    $sub = $value - $mean;
    $medium = $medium + ($sub**2) / $n;
   }
  }
}

$deviation = sqrt($medium);

@output2 = ();

for ($i = 0; $i < @outputs; ++$i) {
  for ($j = 0; $j < $l; ++$j) {
   $value = $outputs[$i][$j];
```

```
    if ($value != 0) {
      $zscore = ($value - $mean) / $deviation;
    }else {
      $zscore = -9;
    }
    $zscore = sprintf("%5.2f", $zscore);
    $output2[$i][$j] = $zscore;
  }
}

open( OUTFILE, ">CDKs.zscore" ) or die( "Cannot open file: $!\n" );


print OUTFILE "The following two matrixs have same z and y axis:\n";
print OUTFILE "The x axis is the list in $listfile.\n";
print OUTFILE "The y axis is the cyclin list in files read (same for every file).\n\n";

print OUTFILE "The first matrix display the values read in (dock value):\n";

   for $bref (@outputs) {

   print OUTFILE "@$bref\n";

   }

print OUTFILE "The second matrix display the corresponding zscore:\n";

  for $bref (@output2) {
    print OUTFILE "@$bref\n";
  }

close ( OUTFILE ) or die ( "Can not close file: $!\n" );

 exit;
```

Script11:

```
#!/usr/bin/perl –w
#ZDOCKscorepanel.pl by Xueping Quan
```

```
# read in the ZDOCK score matrix file, and then convert it to a panel.

use GD;

if ($#ARGV != 0 ) {
  print "This script is to transfer a matrix data to a color panel.";
  print "Read in the file that contain the matrix datas.";
  print "Different level of data will be displayed by different color.\n";
  print "Useage:\n";
  print "$0 [matrix file name]\n";
  die;
}

$matrixfile = $ARGV[0];

open( FILE, "$matrixfile" ) or die "\nCannot open file:$!\n";

@inputs = <FILE>;
chomp @inputs;

close (FILE) or die "\nCannot close file:$!\n";

$j = 0;
@matrixes = ();

foreach $line (@inputs) {
  $matrixes[$j][0]  = substr($line, 0, 5); #column 1-5
  $matrixes[$j][1]  = substr($line, 6, 5); #coulmn 7-11
  $matrixes[$j][2]  = substr($line, 12, 5);
  $matrixes[$j][3]  = substr($line, 18, 5);
  $matrixes[$j][4]  = substr($line, 24, 5);
  $matrixes[$j][5]  = substr($line, 30, 5);
  $matrixes[$j][6]  = substr($line, 36, 5);
  $matrixes[$j][7]  = substr($line, 42, 5);
  $matrixes[$j][8]  = substr($line, 48, 5);
  $matrixes[$j][9]  = substr($line, 54, 5);
  $matrixes[$j][10] = substr($line, 60, 5);
  $matrixes[$j][11] = substr($line, 66, 5);
  $matrixes[$j][12] = substr($line, 72, 5);
  $matrixes[$j][13] = substr($line, 78, 5);
  $matrixes[$j][14] = substr($line, 84, 5);
  $matrixes[$j][15] = substr($line, 90, 5);
  $matrixes[$j][16] = substr($line, 96, 5);
  $matrixes[$j][17] = substr($line, 102, 5);
  $matrixes[$j][18] = substr($line, 108, 5);
```

```
 $matrixes[$j][19] = substr($line, 114, 5);
 $matrixes[$j][20] = substr($line, 120, 5);
 $matrixes[$j][21] = substr($line, 126, 5);
 $matrixes[$j][22] = substr($line, 132, 5);
 $matrixes[$j][23] = substr($line, 138, 5);
 $matrixes[$j][24] = substr($line, 144, 5);
 $matrixes[$j][25] = substr($line, 150, 5);
 $matrixes[$j][26] = substr($line, 156, 5);
 $matrixes[$j][27] = substr($line, 162, 5);
 $matrixes[$j][28] = substr($line, 168, 5);
 $matrixes[$j][29] = substr($line, 174, 5);
 $matrixes[$j][30] = substr($line, 180, 5);
 $matrixes[$j][31] = substr($line, 186, 5);
 $matrixes[$j][32] = substr($line, 192, 5);

 ++$j;
}

for ($j = 0; $j < @matrixes; ++$j) {
 for ($k = 0; $k < 33; ++$k) {
  $matrixes[$j][$k] =~ s/^\s*//g;
  chomp ($matrixes[$j][$k]);
 }
}

@CYCS = ();
$CYCS[0] = 'A01';
$CYCS[1] = 'A02';
$CYCS[2] = 'A03';
$CYCS[3] = 'A04';
$CYCS[4] = 'A05';
$CYCS[5] = 'A06';
$CYCS[6] = 'A07';
$CYCS[7] = 'A08';
$CYCS[8] = 'A09';
$CYCS[9] = 'A10';
$CYCS[10] = 'A11';
$CYCS[11] = 'A12';
$CYCS[12] = 'A13';
$CYCS[13] = 'A14';
$CYCS[14] = 'A15';
$CYCS[15] = 'A16';
$CYCS[16] = 'A17';
$CYCS[17] = 'A18';
$CYCS[18] = 'A19';
```

```
$CYCS[19] = 'A20';
$CYCS[20] = 'A21';
$CYCS[21] = 'A22';
$CYCS[22] = 'A23';
$CYCS[23] = 'A24';
$CYCS[24] = 'A25';
$CYCS[25] = 'A26';
$CYCS[26] = 'A27';
$CYCS[27] = 'A28';
$CYCS[28] = 'A29';
$CYCS[29] = 'A30';
$CYCS[30] = 'A31';
$CYCS[31] = 'A32';
$CYCS[32] = 'A33';
$CYCS[33] = 'A34';
$CYCS[34] = 'A35';
$CYCS[35] = 'A36';


@CDKS = ();
$CDKS[0] = 'B01';
$CDKS[1] = 'B02';
$CDKS[2] = 'B03';
$CDKS[3] = 'B04';
$CDKS[4] = 'B05';
$CDKS[5] = 'B06';
$CDKS[6] = 'B07';
$CDKS[7] = 'B08';
$CDKS[8] = 'B09';
$CDKS[9] = 'B10';
$CDKS[10] = 'B11';
$CDKS[11] = 'B12';
$CDKS[12] = 'B13';
$CDKS[13] = 'B14';
$CDKS[14] = 'B15';
$CDKS[15] = 'B16';
$CDKS[16] = 'B17';
$CDKS[17] = 'B18';
$CDKS[18] = 'B19';
$CDKS[19] = 'B20';
$CDKS[20] = 'B21';
$CDKS[21] = 'B22';
$CDKS[22] = 'B23';
$CDKS[23] = 'B24';
$CDKS[24] = 'B25';
```

```
$CDKS[25] = 'B26';
$CDKS[26] = 'B27';
$CDKS[27] = 'B28';
$CDKS[28] = 'B29';
$CDKS[29] = 'B30';
$CDKS[30] = 'B31';
$CDKS[31] = 'B32';
$CDKS[32] = 'B33';

$image = GD::Image->newPalette (700,760);


$white     = $image->colorAllocate(255,255,255); #value:0
$whiteblue = $image->colorAllocate(170,255,255); #value[40,50)
$cyan      = $image->colorAllocate(0,255,255);   #value[50,60)
$darkcyan  = $image->colorAllocate(0,180,255);   #value[60,70)
$blue      = $image->colorAllocate(0,100,255);   #value[70,80)
$darkblue  = $image->colorAllocate(0,0,240);     #value[80,90]
$black     = $image->colorAllocate(0,0,0);


$image->filledRectangle(0,0,700,760,$white);

for ($i = 0; $i < 36; ++$i) {
  for ($j = 0; $j < 33; ++$j) {
    $x = $j + 1;
    $y = $i + 1;
    $x2 = $j + 2;
    $y2 = $i + 2;
    $value = $matrixes[$i][$j];
  if ($value == 0) {
     $image->filledRectangle($x*20, $y*20, $x2*20, $y2*20, $white);
     $image->rectangle($x*20, $y*20, $x2*20, $y2*20, $black);
   }elsif($value < 50 && $value >= 40) {
     $image->filledRectangle($x*20, $y*20, $x2*20, $y2*20, $whiteblue);
     $image->rectangle($x*20, $y*20, $x2*20, $y2*20, $black);
   }elsif($value >= 50 && $value < 60) {
     $image->filledRectangle($x*20, $y*20, $x2*20, $y2*20, $cyan);
     $image->rectangle($x*20, $y*20, $x2*20, $y2*20, $black);
   }elsif ($value >= 60 && $value < 70) {
     $image->filledRectangle($x*20, $y*20, $x2*20, $y2*20, $darkcyan);
     $image->rectangle($x*20, $y*20, $x2*20, $y2*20, $black);
   }elsif($value >= 70 && $value < 80) {
     $image->filledRectangle($x*20, $y*20, $x2*20, $y2*20, $blue);
     $image->rectangle($x*20, $y*20, $x2*20, $y2*20, $black);
```

```
  }elsif($value >= 80) {
    $image->filledRectangle($x*20, $y*20, $x2*20, $y2*20, $darkblue);
    $image->rectangle($x*20, $y*20, $x2*20, $y2*20, $black);
  }
 }
}

$image->rectangle(20,20,680,740,$black);


for ($i = 0; $i < 36; ++$i) {
$coor = (20 * $i) + 24;
$image->string(gdSmallFont, 2, $coor, "$CYCS[$i]", $black);
}

for ($i = 0; $i < 33; ++$i) {
$coor = (20 * $i) + 21;
$image->string(gdSmallFont, $coor, 8, "$CDKS[$i]", $black);
}


open (OUTFILE, ">pane.png") or die "Cannot open file!";
binmode OUTFILE;
print OUTFILE $image->png;
close OUTFILE;


exit;
```

Script12:

```
#!/usr/bin/perl -w
#polarpercentage.pl by Xueping Quan
# this script is to calculate the polar residue percentage on interface
# it read in the file listing contact files created by PDBcontaceauto.pl
```

```
# then calculate the number of polar residue and total residue numbers
# of each chain in each contact file

# read in the contact file

if ($#ARGV !=0) {
   print "The scripts is to calcualte the polar percentage on interface.";
   print "Read in the contact file list file name";
   print "then calculate polar and total residue numbers for each chain in each file.";
   print "Useage:\n";
   print "$0 [contact file list name]\n";
   die;
}

$contactfile = $ARGV[0];

open ( INPUT, "<$contactfile" ) || die ( "Cannot open file: $!\n" );
@contacts = <INPUT>;
chomp @contacts;
close (INPUT) || die ( "Cannot close file: $!\n" );

foreach $pdb (@contacts) {
  $conname = $pdb . ".con";
  $polarname = $pdb . ".perct";

  open ( INFILE, "<$conname" ) || die ( "Cannot open file: $!\n" );
  @confile = <INFILE>;
  close (INFILE) || die ( "Cannot close file: $!\n" );


#read the file into two arrays.
$i = 0;
$m = 0;
@proteinA = ();
@proteinB = ();

foreach $line (@confile) {
 if ($i > 1) {
    $resID1 = substr($line, 15, 5); #column 16-20
    $res1   = substr($line, 11, 3); #column 12-14
    $res2   = substr($line, 40, 3); #column 41-44
    $resID2 = substr($line, 44, 5); #colimn 44-49

   # put $resID2 and $res2 into one scalar
   $residue1 = $resID1 . $res1;
```

```
  $residue2 = $resID2 . $res2;

  # store information in arrays
  $proteinA[$m] = $residue1;
  $proteinB[$m] = $residue2;
  ++$m;
 }else{
  ++$i;
 }
}

@chainA = sort @proteinA;
@chainB = sort @proteinB;

$totalA = 0;
$polarA = 0;
$totalB = 0;
$polarB = 0;
$polar = 'CYS,ASN,GLN,SER,THR,ARG,HIS,LYS,ASP,GLU';

for ($i = 0; $i < @chainA; ++$i) {
  $temp = $chainA[$i];
  @tempt = split(',$temp);
  splice (@tempt, 0, 5);
  $amino = join(',@tempt);

  if ($i == 0) {
    $totalA = 1;
    if ($polar =~ /$amino/) {
      $polarA = 1;
    }
  }else{
    $temp2 = $chainA[$i - 1];
    if($temp ne $temp2) {
     ++$totalA;
     if ($polar =~ /$amino/) {
       ++$polarA;
     }
    }
  }
}

for ($i = 0; $i < @chainB; ++$i) {
   $temp = $chainB[$i];
   @tempt = split(',$temp);
```

```
  splice (@tempt, 0, 5);
  $amino = join('',@tempt);
  if ($i == 0) {
    $totalB = 1;
    if ($polar =~ /$amino/) {
    $polarB = 1;
    }
  }else{
    $temp2 = $chainB[$i - 1];
    if($temp ne $temp2) {
      ++$totalB;
      if ($polar =~ /$amino/) {
        ++$polarB;
      }
    }
  }
}

$percentageA = sprintf("%6.5f", ($polarA/$totalA));
$percentageB = sprintf("%6.5f", ($polarB/$totalB));

open( OUTFILE, ">$polarname" ) or die( "Cannot open file: $!\n" );

print OUTFILE "The number of polar residues of chain A on interface: $polarA \n";
print OUTFILE "the polar percentage of chain A on interface is $percentageA \n";

print OUTFILE "The number of polar residues of chain B on interface: $polarB \n";
print OUTFILE "the polar percentage of chain B on interface is $percentageB \n";

close (OUTFILE) or die ("Can not close file: $!\n");
}
exit;
```

REFERENCES

1. Abagyan RA and Totrov MM, 1997. Contact area difference (CAD): a robust measure to evaluate accuracy of protein models. *J. Mol. Biol.* 268: 678-685.

2. Acosta JAT, Engler J.de, Raes J, Magyar Z, Groodt RD, Inzé D, Veylder L.De, 2004. Molecular characterization of *Arabidopsis* PHO80-like proteins, a novel class of CDKA;1-interacting cyclins. *Cell Mol Life Sci.* 61:1485-97.

3. Aloy P, Moont G, Gabb HA, Querol E, Aviles FX, Sternberg MJE. 1998. Modelling Protein Docking using Shape Complimentarity, Electrostatics and Biochemical Information. *Proteins: Structure, Function, and Genetics.* 33:535-549. http://www.bmm.icnet.uk/docking/index.html

4. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J. Mol. Boil.* 215:403-410.

5. Andersen G, Busso D, Poterszman A, Hwang JR, Wurtz JM, Ripp R, Thierry JC, Egly JM and Moras D. 1997. The structure of cyclin H: common mode of kinase activation and specific features. *The EMBO J.* 16:958-967.

6. Andrade MA, Ouzounis C, Sander C, Tamames J, Valencia A., 1999. Functional classes in the three domains of life. *J. Mol. Evol.* 49:551-557.

7. *Arabidopsis* Genome Initiative, 2000. Analysis of the genome sequence of the flowering plant *Arabidopsis thaliana. Nature*, 408:796-815.

8. Argos P. 1988. An investigation of protein subunit and domain interfaces. *Protein Eng.* 2:101-113.

9. Bairoch A, Boeckmann B, Ferro S, Gasteiger E, 2004. Swiss-Prot: a juggling between evolution and stability. *Brief Bioinform.* 5:39-55. http://www.swissprot20.org/

10. Barroco RM, De Veylder L, Magyar Z, Engler G, Inze D, Mironov V. 2003. Novel complexes of cyclin-dependent kinases and a cyclin-like protein from Arabidopsis thaliana with a function unrelated to cell division. *Cell Mol Life Sci* 60: 401–412.

11. Baker NA, Sept D, Joseph S, Holst MJ, McCammon JA. 2001. Electrostatics of nanosystems: application to microtubules and the ribosome. *Proc.Natl Acad.Sci. USA* 98: 10037-10041.

12. Bateman A, Coin L, Durbin R, Finn RD, Hollich V, Griffiths-Jones S, Khanna A, Marshall M, Moxon S, Sonnhammer ELL, Studholme DJ, Yeats C and Eddy SR. 2004. The Pfam protein family database. *Nucleic Acids Res.* 32:D138-D141.

13. Bates PA, Kelley LA, MacCallum RM, and Sternberg MJ, 2001. Enhancement of protein modelling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. *Proteins*. 45:39-46.

14. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL, 2004. Genbank: update. *Nucleic Acids Res.* 1;32(database issue): 23-26.

15. Berg J, Willmann S, Lassig M. 2004. Adaptive evolution of transcription factor binding sites. *BMC Evol. Biol.*28:42-53.

16. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. 2000. The Protein Data Bank. *Nucleic Acids Res.* 28: 235-242.

17. Blinn JF, 1982. A Generalization of Algebraic Surface Drawing, *ACM Transactions on Graphics*, 1:235-256.

18. Blundell TL, Sibanda BL, Sternberg MJE, and Thornton JM. 1987. Knowledge-based prediction of protein structures and the design of novel molecules. *Nature*, 326:347-352.

19. Bogan AA and Thorn KS, 1998. Anatomy of Hot Spots in Protein Interfaces. *J.Mol.Biol.* 280:1-9.

20. Brown NR, Noble MEM, Endicott JA, Garman EF, Wakatsuki S., Mitchell E, Rasmussen B., Hunt T., Johnson LN., 1995, the crystal structure of cyclin A. *Structure* 3: 1235-1247.

21. Boudolf V., Rombauts S., Naudts M., Inze D., and De Veylder L., 2001, Identification of novel cyclin-dependent kinases interacting with the CKS1 protein of *Arabidopsis*. *J. Exp. Bot.* 52: 1381-1382.

22. Bower MJ, Cohen FE, Dunbrack RL Jr. 1997. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: A new homology modeling tool. *J. Mol. Biol.*, 267:1268-1282.

23. Bruccoleri RE and Karplus M. 1987. Prediction of the folding of short polypeptide segments by uniform conformational sampling. *Biopolymers*, 26:137-168.

24. Bureesens S., Van Montagu M., and Inze D., 1998, The cell cycle in *Arabidopsis*, *Plant Phys. Biochem.* 36: 9-19.

25. Canutescu AA, Shelenkov AA, Dunbrack Jr RL, 2003. A graph-theory algorithm for rapid protein side-chain prediction. *Protein Science.* 12:2001-2014.

26. Card GL, Knowles P, Laman H, Jones N and McDonal NQ. 2000. Crystal structure of a γ-herpesvirus cyclin-cdk complex. *The EMBO Journal.* 19: 2877-2888.

27. Chakravarty S, Wang L and Sanchez R, 2005. Accuracy of structure-derived properties in simple comparative models of protein structures. *Nucleic Acid Research*: 33: 244-259.

28. Chapman DL, 1913. A contribution to the theory of electrocapillarity. *Phil. Mag.* 25: 475-481.

29. Chen HH, Wang YC, Fan MJ, 2006. Identification and Characterization of the CDK12/Cyclin L1 Complex Involved in Alternative Splicing Regulation. *Molecular and Cellular Biology*. 26: 2736-2745.

30. Chen R, Li L, Weng Z, 2003. ZDOCK: An Initial-stage Protein-Docking Algorithm. *Proteins* 52:80-87. http://zdock.bu.edu/

31. Chen R and Weng ZP, 2002. Docking unbound proteins using shape complementarity, desolvation, and electrostatics. *Proteins: Structure, Function, and Genetics*. 47:281-294.

32. Chothia C. 1974. Hydrophobic bonding and accessible surface area in proteins. *Nature*. 248:338-339.

33. Chothia C, 1976. The nature of the accessible and buried surfaces in proteins. *J. Mol. Biol.* 105, 1-14.

34. Chothia, C. and Janin, J. 1999. Principles of protein-protein recognition. *Nature* 256, 705-708.

35. Chothia C and Lesk AM. 1987. Canonical structures for the hypervariable regions of immunoglobulins. *J. Mol. Biol.*, 196:901-917.

36. Chothia C, Lesk AM, Levitt M, Amit AG, Mariuzza RA, Phillips SEV, and Poljak RJ. 1986. The predicted structure of immunoglobulin d1.3 and its comparison with the crystal structure. *Science*, 233:755-758.

37. Chothia C. and Janin J, 1975. Principles of protein-protein recognition. *Nature.* 256:705-708.

38. Cieplak P, Cornell WD, Bayly C, Kollman PA. 1995. Application of the multimolecule and multiconformational RESP methodology to biopolymers - charge derivation for DNA, RNA and proteins. *J. Comp. Chem.* 16:1357-1377

39. Clackson T and Wells JA, 1995. A hot spot of binding energy in a hormone-receptor interface. *Science.* 267:383-386.

40. Colloc'h N and Mornon JP. 1990. A New Tool for the Qualitative and Quantitative Analysis of Protein Surfaces Using B-Spline and Density of Surface Neighbourhood. *J.Mol.Graphics* 8:133-140.

41. Colovos C, Yeates TO. 1993. Verification of protein structures: patterns of nonbonded atomic interactions. *Protein Sci.* 2:1511-1519

42. Comeau SR, Gatchell DW, Vajda S and Camacho CJ. 2004. ClusPro: a fully automated algorithm for protein–protein docking. *Nucleic Acids Research,* 32:96-99. http://nrc.bu.edu/cluster/

43. Connolly ML, 1983a. Analytical molecular surface calculation. *Journal of Applied Crystallography.* 16:548-558.

44. Connolly ML, 1983b. Solvent-accessible surface of proteins and nucleic acids. *Science*. 221: 709-713.

45. Connolly ML, 1996. Molecular Surfaces: A Review. http://www.netsci.org/Science/Compchem/feature14.html

46. Dandekar T, Snel B, Huynen M and Bork P. 1998. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem Sci* 23:324–328.

47. Davies DR and Cohen GH. 1996. Interactions of protein antigens with antibodies. *Proc Natl Acad Sci USA*. 93: 7–12.

48. Dayhoff MO, Schwartz RM., Orcutt BC. 1978. "A model of evolutionary change in proteins." in "Atlas of Protein Sequence and Structure", National Biomedical Research Foundation, Washington. 345-352.

49. Debondt, H.L., Rosenblatt, J., Jones, H.D., Morgan, D.O. and Kim, S.H., 1993, Crystal structure of cyclin-dependent kinase 2. *Nature*. 363: 595-602.

50. Defalco G. and Giordano A., 1998, CDK9 (PITALRE): a multifunctional cdc2-related kinase. *J. Cell Physiol.* 177: 501-506.

51. de Graaf K, Hekerman P, Spelten O, Herrmann A, Packman LC, Bussow K, Muller-Newen G, Becker W. 2004. Characterization of cyclin L2, a novel cyclin with an arginine/serine-rich domain: phosphorylation by DYRK1A and colocalization with splicing factors. *J Biol Chem.* 279: 4612-4624.

52. Delano WL, Ultsch MH, de Vos AM and Wells JA. 2000. Convergent solutions to binding at a protein-protein interface. *Science*. 287:1279-1283.

53. Deveylder G., Segers G., Glab N., Casteel P., Van Montagu.M. and Inze D., 1997., the *Arabidopsis* Cks1 At protein binds the cyclin-dependent kinases Cdce2aAt and Cdc2bAt. *FEBS lett.* 412: 446-452.

54. Dill KA, Truskett TM, Vlachy V, Hribar-Lee B., 2005. Modeling water, the hydrophobic effect, and ion salvation. *Ann. Rev. Biophys. Biomol. Struct.* 34: 173-199.

55. Dolinsky TJ, Nielsen JE, McCammon JA, Baker NA. 2004. PDB2PQR: an automated pipeline for the setup, execution, and analysis of Poisson-Boltzmann electrostatics calculations. *Nucleic Acids Research* 32:665-667.

56. Dunbrack RL and Karplus M. 1993. Prediction of protein side-chain conformations from a backbone conformation dependent rotamer library. *J. Mol. Biol.* 230:543-571.

57. Eddy SR. 1998. Profile hidden Markov models. *Bioinformatics.* 14:755-763.

58. Eisenberg, D, Weiss RM, Terwilliger TC and Wilcox W. 1982. Hydrophobic Moments and Protein Structure. *Faraday Symp. Chem. Soc.* 17:109-20.

59. Eisenberg, D., Wesson, M., and Yamashita, M. 1989. Interpretation of Protein Folding and Binding with Atomic Solvation Parameters. *Chemica Scripta* 29: 217-221.

60. Enright AJ, Iliopoulos I, Kyrpides NC and Ouzounis CA, 1999. Protein interaction maps for complete genomes based on gene fusion events. *Nature* 402:86–90.

61. Feng DF, Doolittle RF. 1987. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol* 25: 351-60.

62. Fine RM, Wang H, Shenkin PS, Yarmush DL, and Levinthal C. 1986. Predicting antibody hypervariable loop conformations. II: Minimization and molecular dynamics

studies of MCP603 from many randomly generated loop conformations. *Proteins*, 1:342-362.

63. Fischer D, Lin SL, Wolfson HL & Nussinov R, 1995. A geometry based suite of molecular docking processes. *J.Mol.Biol.* 248:459-477.

64. Fischer EE. 1894. der Configuration auf die Wirkung der Enzyme. Ber Dtsch Chem Ges 27: 2984–2993, 1894.

65. Fraga S, Parker JM, Pocok JM. 1995. Computer simulations of protein structures and interactions. New York: Springer Verlag. 2081 p.

66. Fryxell KJ, 1996. The coevolution of gene family trees. *Trends Genet* 12:364–369.

67. Gaasterland T & Ragan MA. 1998. Microbial genescapes: phyletic and functional patterns of ORF distribution among prokaryotes. *Microb Comp Genomics* 3:199–217.

68. Gabdoulline RR, Wade RC & Walther D, 1999. MolSurfer: two dimensional maps for navigating three-dimensional structures of proteins, *Trends Biochem. Sci.*, 24: 285-287.

69. Gibas C. and Jambeck P., 2001. Developing bioinformatics computer skills. O'Reilly & Associates, Inc, 101 Morris Street, Sebastopol, CA 95472.

70. Gilson MK, Sharp KA, Honig B, 1987. Calculating the electrostatic potential of molecules in solution: method and error assessment. *J. Comp. Chem.* 9:327-335.

71. Goh C-S, Bogan AA, Joachimiak M, Walther D and Cohen FE, 2000. Co-evolution of proteins with their interaction partners. *J Mol Biol* 299:283–293.

72. Gonnett GH, Hallett MT, Korostensky C, and Bernardin L. 2000. Darwin v. 2.0: an interpreted computer language for the biosciences. *Bioinformatics*. 16: 101-103.

73. Gough J, Hughey R, Karplus K, and Chothia C. 2001. Assignment of genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J Mol Biol*. 313:903-919.

74. Gouy M, 1910. Sur la. Constitution de la change electrique a la surface d'un electrolyte. *Journ de. Phys*. 9:457-468.

75. Granas X, Claudio PP, De Luca A, Sang N, Giordano A, 1994. PISSLRE, a human novel CDC2-related protein kinase. *Oncogene*, 9:2097-2103.

76. Gray JJ, Moughon S, Wang C, Kuhlman OSFB, Rohl CA, Baker D. 2003. Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. *J Mol Biol*. 331:281-299.

77. Greer J, 1990. Comparative modelling methods: application to the family of the mammalian serine protease. *Proteins*. 7:317-334.

78. Grundy, William N., Timothy L. Bailey, Charles P. Elkan and Michael E. Baker, 1997. Meta-MEME: Motif-based Hidden Markov Models of Biological Sequences. *Computer Applications in the Biosciences*, 13:397-406.

79. Gupta SK, Kececioglu JD & Schäffer AA, 1995. Improving the Practical Time and Space Efficiency of the Shortest-Paths Approach to Sum-of-Pairs Multiple Sequence Alignment. *J. Computational Biology* 2:459-472.

80. Halperin I, Ma BY, Wolfson H, and Nussinov R, 2002. Principles of docking: an overview of search algorithms and a guide to scoring functions. *Proteins*. 47:409-443.

81. Hanks, S.K., Hunter, T., 1995, Protein kinases 6. The eukaryotic protein kinase superfamily: kinase (catalytic) domain structure and classification. *FASEB J.* 19: 576-596.

82. Hendsch ZS & Tidor B. 1994. Do salt bridges stabilize proteins? A continuum electrostatic analysis. *Protein Science*, 3:211-226.

83. Honig B & Hubbell WL, 1984. Stability of "salt bridges" in membrane proteins. *Proc.Natl.Acad.Sci.USA*. 81:5412-5416.

84. Henikoff S, Henikoff JG, 1992. Amino acid substitution matrices from protein blocks. *Proc.Natl.Acad.Sci. USA* 89:10915-10919.

85. Hertz-Fowler C, Peacock CS, Wood V, Aslett M, Kerhornou A, Mooney P, Tivey A, Berriman M, Hall N, Rutherford K, Parkhill J, Ivens AC, Rajandream MA, and Barrell B. 2004. GeneDB: a resource for prokaryotic and eukaryotic organisms. *Nucleic Acids Research.* 32: D339-D343. http://www.genedb.org/

86. Hooft RWW, Sander C, and Vriend G. 1996. Verification of protein structures: side-chain planarity. *J. Appl. Crystallog.*, 29:714-716.

87. Horton N & Lewis M, 1992. Calculation of the free energy of association for protein complexes. *Protein Sci.* 1:169-181.

88. Horwich AL and Weissman JS, 1997. Deadly Conformations: Protein Misfolding in Prion Disease. *Cell.* 89: 499-510.

89. Hubbard SJ & Argos P, 1994. Cavities and packing at protein interfaces. *Prot. Sci.* 3: 2194-2206.

90. Hughey R, Krogh A. 1996. Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput Appl Biosci.* 12:95-107.

91. Huse M. and Kuriyan J., 2002. The conformational plasticity of protein kinases. *Cell.* 109:275-282.

92. Janin, J. 2002. Welcome to CAPRI: a critical assessment of predicted interactions. *Proteins.* 47:257.

93. Janin J, 2005. Assessing predictions of protein-protein interactions: the CAPRI experiment. *Protein Science.* 14:278-283.

94. Janin J, Miller S, Chothis C, 1988. Surface, subunit interfaces and interior of oligomeric proteins. *J.Mol.Biol.*, 204:155-164.

95. Jeffrey P.D., Russo A.A., Polyak K., Gibbs E., Hurwitz J., Massague J. & Pavletich N.P., 1995. Mechanism of CDK activation revealed by the structure of a cyclinA-CDK2 complex. *Nature.* 376:313-320.

96. Jeffrey PD, Tong LL and Pavletich NP. 2000. Structural basis of inhibition of CDK-cyclin complexes by INK4 inhibitors. *Genes and Developments.* 14:3115-3125.

97. Jensen MO, Mouritsen OG, Peters GH, 2004. The hydrophobic effect: molecular dynamics simulations of water confined between extended hydrophobic and hydrophilic surfaces. *J. Chem. Phys.* 120:9729-9744.

98. Johnson L.N., 2002. Online Laboratory Report. (http://biop.ox.ac.uk/www/lj2001/preface.html)

99. Jones S. and Thornton JM., 1995. Protein-Protein Interactions: A Review of Protein Dimer Structures. *Progress in Biophysics and Molecular Biology*: 63:p31-165.

100. Jones S and Thornton JM, 1996. Principles of protein-protein interactions. *Proc Natl Acad Sci USA*, 93:13-20.

101. Jones S and Thornton JM, 1997. Analysis of protein-protein interaction sites using surface patches. *J.Mol.Biol.* 272:121-132.

102. Jones TH and Thirup S. 1986. Using known substructures in protein model building and crystallography. *EMBO J.*, 5:819-822.

103. Joubès J., Chevalier C., Dudits D., Heberle-Bors E., Inzé D., Umeda M. and Renaudin J.P., 2000, CDK-related protein kinases in plants. *Plant Mol. Bio.* 43: 607-620.

104. Karlin, S, and Altschul SF, 1993. Applications and statistics for multiple high-scoring segments in molecular sequences. *Proc. Natl. Acad. Sci.* 90:5873-7.

105. Karplus K, Barrett C, Hughey R, 1998. Hidden Markov Models for Detecting Remote Protein Homologies. *Bioinformatics*. 14:846-856.

106. Katchalski-Katzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA, 1992. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc Natl Acad Sci U S A*. 89:2195-2199.

107. Kayoko Yamada, Jun Lim, Joseph M. Dale, Huaming Chen, Paul Shinn, Curtis J. Palm, Audrey M. Southwick, Hank C. Wu, Christopher Kim, Michelle Nguyen, Paul Pham, Rosa Cheuk, George Karlin-Newmann, Shirley X. Liu, Bao Lam, Hitomi Sakano, Troy Wu, Guixia Yu, Molly Miranda, Hong L. Quach, Matthew Tripp, Charlie H. Chang, Jeong M. Lee, Mitsue Toriumi, Marie M. H. Chan, Carolyn C. Tang, Courtney S. Onodera, Justine M. Deng, Kenji Akiyama, Yasser Ansari, Takahiro Arakawa, Jenny Banh, Fumika Banno, Leah Bowser, Shelise Brooks, Piero Carninci, Qimin Chao, Nathan Choy, Akiko Enju, Andrew D. Goldsmith, Mani Gurjal, Nancy F. Hansen, Yoshihide Hayashizaki, Chanda Johnson-Hopson, Vickie W. Hsuan, Kei Iida, Meagan Karnes, Shehnaz Khan, Eric Koesema, Junko Ishida, Paul X. Jiang, Ted Jones, Jun

Kawai, Asako Kamiya, Cristina Meyers, Maiko Nakajima, Mari Narusaka, Motoaki Seki, Tetsuya Sakurai, Masakazu Satou, Racquel Tamse, Maria Vaysberg, Erika K. Wallender, Cecilia Wong, Yuki Yamamura, Shiaulou Yuan, Kazuo Shinozaki, Ronald W. Davis, Athanasios Theologis, and Joseph R. Ecker. 2003. Empirical Analysis of Transcriptional Activity in the Arabidopsis Genome. *Science* 302: 842-846.

108. Kendrew JC, Bodo G, Dintzis HM, Parrish RG, Wyckoff H, Phillips DC. 1958. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature.* 181:662-6.

109. Kikuno R, Nagase T, Ishikawa K, Hirosawa M, Miyajima N, Tanaka A, Kotani H, Nomura N, Ohara O. 1999. Prediction of the coding sequences of unidentified human genes. XIV. The complete sequences of 100 new cDNA clones from brain which code for large proteins in vitro. *DNA Res.* 6: 197-205.

110. Kim KK, Chamberlin HM, Morgan DO and Kim SH, 1996. Three-dimensional structure of human cyclin H, a positive regulator of the CDK-activating kinase. *Nature Structural Biology.* 3:849-855.

111. Kleanthous C, 2000. Protein-Protein Recognition. Oxford University Press.

112. Klotz IM, Darnall DW, Langerman NR, eds. 1975. Quaternary Structure of Proteins. pp. 293411. New York: Academic.

113. Kobayashi H., Stewart E. Poon R, Adamczewick JP. Gannon J., Hunt T., 1992, Identification of the domains in cyclin A required for the binding to, and activation of, $p34^{cdc2}$ and $p34^{cdk2}$ protein kinase subunit. *Mol Biol Cell* 3: 1279-1294.

114. Koehl P and Delarue M, 1994. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J.Mol.Biol.* 239:249-275.

115. Kolch W., 2000. Meaningful relationships: the regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. *Biochem J.* 351:289-305.

116. Kong M, Barnes FA, Ollendorff V, Donoghue DJ, 2000. Cyclin F regulates the nuclear localization of cyclin B1 through a cyclin-cyclin interaction. *EMBO J*, 15, 19(6):1378-1388.

117. Korn AP and Burnett RM, 1991. Distribution and complementarity of hydropathy in multisubunit proteins. *Proteins: Structure, Function Genetics*. 9:37-55.

118. Koshland DE Jr. 1958. Application of a theory of enzyme specificity to protein synthesis. *Proc Natl Acad Sci USA* 44: 98–104.

119. Kraulis PJ, 1991. MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures, *Journal of Applied Crystallography* 24:946-950.

120. Krogh A, Brown M, Mian IS, Sjolander K, Haussler D. 1994. Hidden Markov models in computational biology: Applications to protein modeling. *J. Mol. Biol.*, 235:1501-1531.

121. Kryshtafovych A, Venclovas C, Fidelis K, Moult J. 2005. Progress over the first decade of CASP experiments. *Proteins.* 61 Suppl 7:225-36.

122. Kuntz ID, Blaney JM, Oatley SJ, Langridge R, and Ferrin TE, 1982. A Geometric Approach to Macromolecule-Ligand Interactions. *J. Mol. Biol.* 161:269-288.

123. Kuntz ID, 1992. Structure-Based Strategies for Drug Design and Discovery. *Science*, 257:1078-1082.

124. Lapidot -Lifson Y., Patinkin D., prody C.A., Ehrlich G., Seidman S., Ben-Aziz R., Benseler F., Eckstein F., Zakut H. and Soreq H., 1992, Cloning and antisense oligodeoxynucleotide inhibition of a human homology of cdc2 required in hematooiesis. *Proc. Natl. Acad. Sci. USA* 89: 579-583.

125. Laskowski RA, 1991. SURFNET computer program (Department of Biochemistry and molecular biology, University College, London, England).

126. Laskoswki RA, MacArthur MW, Moss DS & Thorton JM. 1993. PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.* 26, 283-291.

127. Laskowski RA, Luscombe NM, Swindells MB, Thornton JM. 1996. Protein clefts in molecular recognition and function. *Protein.Sci.* 5:2438-2452.

128. Lawrence CE, Altschul SF, Boguski MS, Liu JS, Neuwald AF, Wootton JC. 1993. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science.* 262: 208-214.

129. Lawrence MC and Coleman PM. 1993. Shape complementarity at protein/protein interfaces. *J.Mol.Biol.* 234:946-950.

130. Lee B and Richards FM, 1971. The interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.* 55:379-400.

131. Lees EM., Harlow E., 1993, Sequences within the conserved cyclin box of human cyclin A are sufficient for binding to and activation of cdc2 kinase. *Mol Cell Biol* 13: 1194-1294.

132. Levitt M, 1992. Accurate modeling of protein conformation by automatic segment matching. *J.Mol.Biol.* 226:507-533.

133. Lichtarge O, bourne HR and Cohen FE, 1996. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.* 257:342-358.

134. Linder AB, Eshhar Z, Tawfik DS. 1999. Conformational changes affect binding and catalysis by ester-hydrolysing antibodies. *J.M.Biol.* 285:421-430.

135. Lipman DJ, Altschul S & Kececioglu JD, 1989. A Tool for Multiple Sequence Alignment. *Proc. Natl. Acad. Sci. USA* 86: 4412-4415.

136. Lipman DJ, Pearson WR, 1985. Rapid and sensitive protein similarity searches. *Science*. 227:1435-1441.

137. Littler SJ and Hubbard SJ, 2005. Conservation of orientation and sequence in protein domain-domain interactions. *J.Mol.Biol.*, 345:1265-1279.

138. Ludwig M, Bergman C, Patel N, Kreitman M. 2000. Evidence for stabilizing selection in a eukaryotic enhancer element. *Nature.* 403:564–567

139. Luthy R, Bowie JU, Eisenberg D. 1992. Assessment of protein models with three-dimensional profiles. *Nature* 356: 83-85.

140. Madura JD, Briggs JM, Wade R, Davis ME, Luty BA, Ilin A, Antosiewicz A, Gilson MK, Bagheri B, Ridgway L, McCamm GA, 1995. Electrostatic and diffusion of molecule in solution: simulations with the University of Houston Brownian dynamic program. *Comp. Comm. Phys.* 91:57-95.

141. Magyar Z., Mesazar T., Miskolczi P., Deak M., Feher A., Brown S., Kondorosi E., Athanasisdis A., Pongor S., Bilgin S., Bako L., Koncz C. and Dudits D., 1997, Cell cycle phase sepcificity of putative cyclin-dependent kinase variants in synchronized alfalfa cells. *Plant Cell*, 9:223-235.

142. Mandell JG, Roberts VA, Pique ME, Kotlovyi V, Mitchell JC, Nelson E, Tsigelny Ten Eyck LYF, 2001. Protein docking using continuum electrostatics and geometric fit. *Protein Engineering*. 14: 105-113.

143. Manning G., Whyte D.B., Martinez R., Hunter T., Sudarsanam S., 2002, The protein kinase complement of the human Genome. *Science*, 298: 1912-1934.

144. Marcotte EM, Pellegrini M, Ho-Leung N, Rice DW, Yeates TO and Eisenberg D. 1999. Detecting protein function and protein-protein interactions from genome sequences. *Science* 285:751–753.

145. Martin ACR, Cheetham JC, and Rees AR. 1989 Modeling antibody hypervariable loops: a combined algorithm. *Proc. Natl. Acad. Sci. USA*, 86:9268-9272.

146. MartÃ-Renom MA, Yerkovich B, and Sali A. 2002. Comparative protein structure prediction. *Curr. Protocol. Prot. Sci.* 1: 2.9.1-2.9.22

147. Meijer, L., Jezequel A, Ducommun B (editor), 2000, Progress in Cell Cycle Research, Vol. 4, Plenum Press, New York, 248 pp (21 chapters).

148. Meikrantz W. & Schlegel R. 1996, Suppression of apoptosis by dominant negative mutants of cyclin-dependent protein kinases. *J.Biol.Chem.* 271, 10205-10209.

149. Meng EC, Shoichet BK, and Kuntz ID, 1992. Automated Docking with Grid-Based Energy Evaluation. *J. Comput. Chem.* 13:505-524.

150. Menges M, de Jager SM, Grussiem W, and Murray JAH. 2005. Global analysis of the core cell cycle regulators of Arabidopsis identifies novel genes, reveals multiple and highly specific profiles of expression and provides a coherent model for plant cell cycle control. *The Plant J.* 41:546-566.

151. Meyerson M., Eners GH., Wu CL, Su LK, Gorka C., Nelson C, Harlow E, Tsai LH (1992) A family of human cdc2-related protein kinases. *EMBO J* 11: 2909-2917.

152. Mironov V., Deveylder L., Van Montagu m., and Inze D., 1999, Cyclin-dependent kinases: engine, clocks, and microprocessors. *Annu, Rev. Cell. Dev. Biol.* 13: 261-291.

153. Moont G, Gabb HA, Sternberg MJE, 1999. Use of Pair Potentials Across Protein Interfaces in Screening Predicted Docked Complexes. *Proteins: Structure, Function, and Genetics* 35:364-373. http://www.bmm.icnet.uk/docking/index.html

154. Morgan, D., 1997, Cyclin-dependent kinases: engines, clocks, and microprocessors. *Annu Rev Cell Dev Biol* 13: 261-291.

155. Moult J, Fidelis K, Zemla A, and Hubbard T, 2003. Critical assessment of methods of protein structure prediction (CASP)-round V. *Proteins*. 53:334-339.

156. Moult J and James MNG. 1986. An algorithm for determining the conformation of polypeptide segments in proteins by systematic search. *Proteins,* 1:146-163.

157. Murzin AG, Brenner SE, Hubbard T, Chothia C. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.* 247, 536-540.

158. Nakagami H., Sekine M., Murakami H. and shinmyo A., 1999, tobacco retinoblastoma-related protein phosphorylated by a distinct cyclin-dependent kinase complex with Cdc2/cyclinD in vitro. *Plant J.* 18: 243-252.

159. Naula C, Parsons M, Mottram JC, 2005. Protein Kinases as drug targets in trypanosomes and *leishmania. Biochimica et Biophysica Acta*, 1754:151-159.

160. Needleman SB and Wunsch CD, 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol.* 48:443-453.

161. Nicholls A, Sharp KA and Honig B, 1991. Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons. *Proteins.* 11: 281-296.

162. Nooren IMA and Thornton JM, 2003. Diversity of protein-protein interactions. *The EMBO J.* 22:3486-3492.

163. Nooren IMA and Thornton JM, 2003. Structural characterisation and functional significance of transient protein-protein interactions. *J.Mol.Biol.* 325:991-1018.

164. Norel R, Lin SL, Wolfson HJ & Nussinov R, 1994. Shape complementarity at protein-protein interfaces. *Biopolymers* 34: 933-940.

165. Norel R, Petrey D, Wolfson H, Nussinov R. 1999, Examination of shape complementarity in docking of unbound proteins. *Proteins.* 35:403-419.

166. Notredame C and Higgins DG, 1996. Sequence alignment by genetic algorithm. *Nucleic Acids Res* 24: 1515-24.

167. Notredame C, Higgins DG and Heringa J, 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J.Mol.Biol.* 302:205-217.

168. Nugent JHA, Alfa CE, Young T, Hyams JS, 1991, Conserved structural motifs in cyclins identified by sequence analysis, *J. Cell Sci* 99:669-674.

169. Nurse, P. 1997. Checkpoint pathways come of age. *Cell.* 91:865-867.

170. O'Farrell P. Leopold P, 1991, A consensus of cyclin sequences reveals homology with the ras oncogene. *Cold Spring Harbor Symp Quant boil* 56: 83-92.

171. Ofran Y. and Rost Burkhard, 2003. Analysing six types of protein-protein interfaces. *J.Mol.Biol.* 325:377-387.

172. Orengo CA, Jones DT & Thornton JM. 2003. Bioinformatics: Genes, Proteins & Computers. BIOS Press.

173. O'Sullivan O, Suhre K, Abergel C, Higgins DG, Notredame C. 2004. 3DCoffee: Combining Protein Sequences and Structures within Multiple Sequence Alignments. *J. Mol. Biol.* 340: 385–395. http://www.igs.cnrs-mrs.fr/Tcoffee/tcoffee_cgi/index.cgi?stage1=1&daction=EXPRESSO(3DCoffee)::Regular&config_infile=/home/igs/public_html/Tcoffee/tcoffee_cgi/configuration_file.txt

174. Overbeek R, Fonstein M, D'Souza M, Pusch GD and Maltsev N. 1999. Use of contiguity on the chromosome to predict functional coupling. *InSilico Biol* 1:93–108.

175. Pagano M, Pepperkok R, Verde F, Ansorge W, and Draetta G. 1992. Cyclin A is required at two points in the human cell cycle. *EMBO. J.* 11: 961-971.

176. Pages S, Belaich A, Belaich JP, Morag E, Lamed R, Shoham Y and Bayer EA, 1997. Species-specificity of the cohesin-dockerin interaction between *Clostridium thermocellum* and *Clostridium cellulolyticum*: prediction of specificity determinants of the dockerin domain. *Proteins* 29:517–527.

177. Palma PN, Krippahl L, Wampler JE, Moura JJG. 2000. BiGGER: a new (soft) docking algorithm for predicting protein interactions. *Proteins.* 39:372-84.

178. Park J, Lappe M, Teichmann SA., 2001. Mapping protein family interactions: intramolecular and intermolecular protein family interaction repertoires in the PDB and Yeast. *J.Mol.Biol.* 307:929-938.

179. Pauling L, 1939. The nature of chemical bond. Cornell University Press.

180. Pazos F and Valencia A, 2001. Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Eng* 14:609–614.

181. Pearson WR, Lipman DJ, 1988. Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. USA*. 85:2444-2448.

182. Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D and Yeates TO. 1999. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci USA* 96:4285–4288.

183. Peri S, *et al*, 2003. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research.* 13:2363-2371. http://www.hprd.org/

184. Petrey D, Xiang Z, Tang CL, Xie L, Gimpelev M, Mitros T, Soto CS, Goldsmith-fischman S, Kernytaky A, Schlessinger A, *et al.*, 2003. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modelling. *Proteins*. 53(Suppl. 6):430-435.

185. Pontius J, Richelle J, Wodak SJ. 1996. Deviations from standard atomic volumes as a quality measure for protein crystal structures. *J. Mol. Biol.* 264:121-136.

186. Porceddu A., Stals H., Reichheld J.P., segers G., De Veylder L., De Pinho Barroco R., Casteels P., Van Montagu M., Inze D., and Mironov V., 2001, A plant specific cyclin dependent kinase is involved in the control of G2/M progression in plants. *J. Biol. Chem.* 276: 36354-36360.

187. Renaudin JP, Doonan JH, Freeman D, Hashimoto J, Hirt H, Inzé D, Jacob T, Kouchi H, Rouzé P, Sauter M, Savouré A, Sorrell DA, Sundaresan V, and Murray JAH.

1996. Plant cyclins: a unified nomenclature for plant A-, B- and D-type cyclins based on sequence organization. *Plant Mol. Biol.* 32:1003-1018.

188. Richards FM. 1977. Areas, volumes, packing and protein structure. *Annu. Rev. Biophys. Bioeng.* 6: 151-176.

189. Richards RM, 1974, The interpretation of protein structures: Total volume, group volume distributions and packing density. *J. Mol. Biol.* 82: 1-14.

190. Sayle RA & Milner-White EJ, 1995. RASMOL: biomolecular graphics for all. *Trends Biochem Sci.* 20:374.

191. Richmond TJ and Richards FM. 1978. Packing of a-helices: Geometrical constraints and contact areas. *J. Mol. Biol.* 119: 537-555.

192. Ritchie DW & Kemp GJL, 2000. Protein Docking using spherical polar Fourier correlations. Proteins: Struct. Funct. Genet. 52(1):98-106. http://www.csd.abdn.ac.uk/hex/

193. Saitou N and Nei M, 1987. The neighbor-joining method: A new method for reconstructing phylogenetic tree. *Mol. Biol. Evol.* 4: 406-425.

194. Sali A & Blundell TL, 1993. Comparative protein modelling by satisfaction of spatial restraints. *J. Mol. Biol.* 234:779-815.

195. Sánchez R and Sali A. 1998. Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. *Proc. Natl. Acad. Sci. USA*, 95:13597-13602.

196. Sander C and Schneider R, 1991. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins*. 9:56-68.

197. Schuler GD, Altschul SF, and Lipman DJ. 1991. A Workbench for Multiple Alignment Construction and Analysis. *Proteins* 9: 180-190.

198. Schwede T, Kopp J, Guex N, and Peitsch MC. 2004. SWISS-MODEL: an automated protein homology modelling server. *Nucleic Acids Res.* 31:3381-3385.

199. Sharp KA and Honig B, 1990. Calculation total electrostatic energies with the non-linear Poisson-Boltzmann equation. *J. Phys. Chem.* 94:7684-7692.

200. Shatsky M., Nussinov R. and Wolfson HJ, 2002, MultiProt - a Multiple Protein Structural Alignment Algorithm. *Lecture Notes in Computer Science* 2452:235--250, Springer Verlag.

201. Shih HH, Brady J and Karplus M, 1985. Structure of proteins with single-site mutations: a minimum perturbation approach. *Proc Natl Acad Sci U S A.* 82: 1697–1700.

202. Shindyalov IN, Bourne PE. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering.* 11: 739-747. http://cl.sdsc.edu/

203. Shoichet BK, Bodian DL, and Kuntz ID, 1992. Molecular Docking Using Shape Descriptors. *J. Comput. Chem.* 13:380-397.

204. Singh J. and Thornton J.M., 1991, Atlas of protein-protein side-chain interactions. Oxford University Press.

205. Smith GS and Sternberg MJE, 2002. Prediction of protein-protein interactions by molecular docking methods. *Current Opinion in Structural Biology.* 12: 28-35.

206. Smith TF, Waterman MS, 1981. Identification of Common Molecular Subsequences. *J. Mol. Biol.*, 147:195-197.

207. Smith TF, Waterman MS, 1981 (b). Comparison of biosequences. *Adv. Appl. Math.*, 2, 482–489.

208. Snow M, Amzel LM, 1986. Calculating three-dimensional changes in protein structure due to amino-acid substitutions: the variable region of immunoglobulins. *Proteins.* 1:267-279.

209. Sprinzak E and Margalit H, 2001. Correlated sequence-signatures as markers of protein-protein interactions. *J Mol Biol* 311:681–692.

210. States, DJ, and Gish W, 1994. Combined use of sequence similarity and codon bias for coding region identification. *J. Comp. Biol.* 1:39-50.

211. Sturrock, S. & Collins, J. 1993. MPsrch version 1.3. Biocomputing Research Unit, University of Edinburgh, UK. http://www.ebi.ac.uk/MPsrch/

212. Sutcliffe MJ, Haneef I, Carney D, and Blundell TL. 1987. Knowledge based modelling of homologous proteins, Part I: Three dimensional frameworks derived from the simultaneous superposition of multiple structures. *Protein Eng.*, 1:377-384.

213. Tamames J, Casari G, Ouzounis C and Valencia A. 1997. Conserved clusters of functionally related genes in two bacterial genomes. *J. Mol. Evol.* 44:66–73.

214. Tarricone C, Dhavan R, Peng J, Areces LB, Tsai LH, Musacchio A. 2001. Structure and Regulation of the Cdk5-P25^{Nck5A} Complex. *Mol. Cell* 8:657-669.

215. Tatusov PL, Koonin EV, Lipman DJ, 1997. A genomic perspective on protein families. *Science.* 278:631-637.

216. Thompson JD, Higgins, D.G. and Gibson, T.J. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Res.*, 22:4673-4680.

217. Tobi D & Bahar I, 2005. Structural changes involved in protein binding correlate with intrinsic motions of proteins in the unbound state. *Biophysics*. 52: 18908-18913.

218. Torres Acosta JA, de Almeida Engler J, Raes J, Magyar Z, De Groodt R, Inze D, De Veylder L. 2004. Molecular characterization of Arabidopsis PHO80-like proteins, a novel class of CDKA;1-interacting cyclins. *Cell Mol Life Sci.* 61:1485-1497.

219. Tramontano A, Leplae R, and Morea V, 2001. Analysis and assessment of comparative modelling predictions in CASP4. *Proteins* 45(suppl.5): 22-38.

220. Tramontano A and Morea V, 2003. Assessment of homology-based predictions in CASP5. *Proteins*. 53(suppl.6): 352-368.

221. Tsai CJ, Kumar S, Ma B, Nussinov R. 1999. Folding funnels, binding funnels, and protein function. *Protein Sci.* 8:1181-1190.

222. Tsoka S and Ouzounis CA, 2000. Prediction of protein interactions: metabolic enzymes are frequently involved in gene fusion. *Nat Genet* 26:141–142.

223. Utez P. & Vollert CS, Protein-Protein Interactions. (Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine). http://igtmv1.fzk.de/www/itg/uetz/publications/Uetz2003-PPI.pdf.

224. Vakser IA, 1995. Protein docking for low-resolution structures. *Protein Eng.* 8: 371-377.

225. Vandepoele K., Raes J., Veylder L.D., Rouze P., Rombauts S., and Inze D., 2002, Genome-wide Analysis of Core Cell Cycle Genes in *Arabidopsis*. *Plant Cell*, 14: 903-916.

226. Van Vlijmen HWT and Karplus M. 1997. PDB-based protein loop prediction: Parameters for selection and methods for optimization. *J. Mol. Biol.*, 267:975-1001.

227. Vásquez M. Modeling side-chain conformation. *Curr. Opin. Str. Biol.*, 6:217-221, 1996.

228. Venclovas C , Zemla A, Fidelis K, Moult J. 2003. Assessment of progress over the CASP experiments. *Proteins.* 53 Suppl 6:585-95.

229. Wadman I, Li J, Bash RO, Forster A, Osada H, Rabbitts TH, Baer R. 1994. Specific in vivo association between the bHLH and LIM proteins implicated in human T cell leukemia. EMBO.J. 13:4831-4839.

230. Wakeham N., Terzyan S., Zhai P., Loy J.A., Tang J. and Zhang X. C., 2002. Effects of deletion of streptokinase residues 48-59 on plasminogen activation. Protein Engineering. 15(9): 753-761.

231. Walker JE, Sarast M, Runswick MJ and Gay NJ, 1982. Distantly related sequences in the alpha- and beta-subunits of ATP synthase, myosin, kinases and other ATP-requiring enzymes and a common nucleotide binding fold. *EMBO J.*, 1, 945–951.

232. Wallner B and Elpfsson A, 2005. All are not equal: a benckmark of different homology modelling programs. *Protein Science.* 14:1315-1327.

233. Walls PH and Sternberg MJ. 1992. New algorithm to model protein-protein recognition based on surface complementarity. Applications to antibody-antigen docking. *J Mol Biol.* 228:277-297.

234. Wang G, Kong H, Sun Y, Zhang X, Zhang W, Altman N, dePamphilis CW and Ma H. 2004. Genome-wide analysis of the cyclin family in Arabidopsis and comparative phylogenetic analysis of plant cyclin-like proteins. *Plant Physiol.* 135:1084-1099.

235. Watanabe K and Harayama S, 2001. Swiss-Prot: the curated protein database on internet (in Japnaese). *Protein, Nuclei Acid and Enzyme.* 46:80-86.

236. Yamaguchi M., Fabian T., Sauter M., Bhalerao R.P., Schrader J., Sandberg G., Umeda M.and Uchimiya H., 2000. Activation of CDK-activationg kinase is dependent on interaction with H-type cyclins in plants. *Plant J.*, 24(1): 11-20.

237. Yamaguchi M., Umeda M., and Uchimiya H., 1998, A rice homolog of CDK7/MO15 phosphorylates both cyclin-dependent protein kinases and the carboxy-terminal domain of RNA polymerase II. *Plant J.* 16: 613-619.

238. Zdobnov EM, Lopez R, Apweiler R, and Etzold T. 2002. The EBI SRS Server-Recent Developments. *Bioinformatics*. 18: 368-373.

239. Zhan Q, Antinore MJ, Wang XW, Carrier F, Smith ML, Harris CC, Fornace AJ Jr. 1999. Association with Cdc2 and inhibition of Cdc2/cyclin B1 kinase activity by the p53-regulated protein Gadd45.Oncogene, 18:2892-2900. http://srs.ebi.ac.uk/srsbin/cgi-bin/wgetz?-page+srsq2+-noSession

240. Zvelebil MJ, Barton GJ, Taylor WR & Sternberg MJ, 1987. Prediction of protein secondary structure and active sites using the alignment of homologous sequences. *J.Mol.Biol.* 195:957-961.